Altera SoC Linux Intro Workshop

Altera SW SoC Workshop Series

- SW Workshop #1 – Altera SoC SW Development Overview
- SW Workshop #2 – Introduction to Linux on Altera SoC
- SW Workshop #3 – Developing Drivers for Altera SoC Linux

Agenda

- Essential Information Resources
- SoC Device Overview
- SoC Physical Address Map
- SoCFPGA Development Flow & Tools
- Altera SoC Linux Overview
- Components of the SoC FPGA Linux BSP
- SoC Linux Upstreaming & Driver Support
- Altera SoC Linux Boot Flow
- Das U-Boot Bootloader
- Linux Device Tree for SoC FPGA
- Take Home Lab

Welcome. Here’s What You Can Expect Today

Experienced Linux Developers

- Find a familiar embedded Linux development flow
- Overview of upstreaming and driver support for mach_SoCFPGA architecture
- Guide to SoCFPGA resources

New Linux Developers

- An exposure to the components of embedded Linux
- Essential Linux learning and documentation resources

Hardware Developers

- HW handoff to Linux build flow
- Boot and FPGA configuration for Linux
- SW implications of HW architecture

Everyone

- SoC FPGA architecture-specific information
- SoC FPGA recommendations and best practices

Focused on SoC/Nios Linux Specific Topics

Essential Information Resources

Where to learn more…

…a non-exhaustive list

Linux Foundation Training

Linux Developer classes are designed to help participants:

- Learn how to develop an embedded Linux product
- Become familiar with and learn to write device drivers
- Get practical experience with the Linux kernel
- Learn how to work with the Linux developer community

Developer Courses

- *LFD331* – Developing Linux Device Drivers
- *LFD405* – Building Embedded Linux with the Yocto Project
- *LFD411* – Embedded Linux Development
- *LFD414* – Introduction to Embedded Android Development
- *LFD205* – How to Participate with the Linux Community
- *LFD211* – Introduction to Linux for Developers
- *LFD262* – Developing with Git
- *LFD312* – Developing Applications for Linux
- *LFD320* – Linux Kernel Internals & Debugging
- *LFD415* – Inside Android: An Intro to Android Internals
- *LFD432* – Optimizing Linux Device Drivers for Power Efficiency

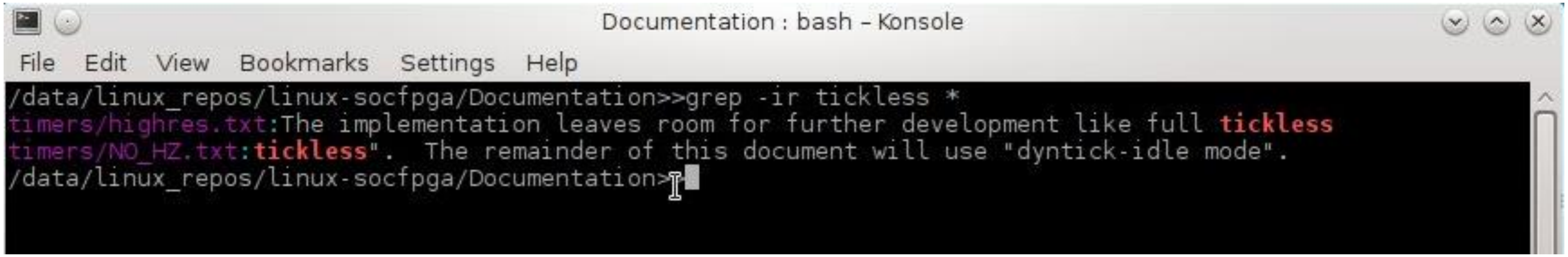
http://training.linuxfoundation.org/linux-courses/development-training

Linux Documentation Resources

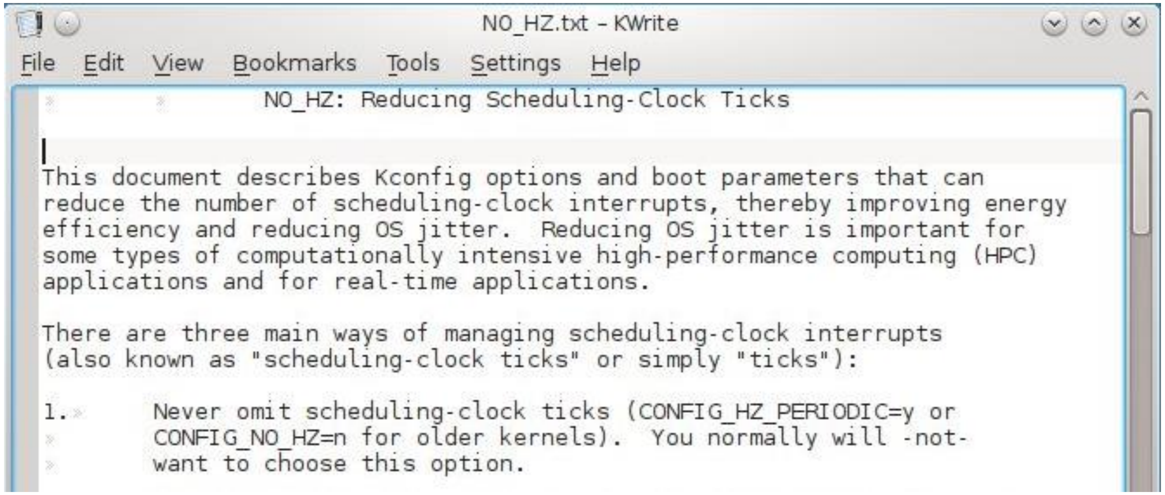
- GIT
  - Distributed revision control system to enable distributed collaboration
  - On-line documentation & training:
    - http://git-scm.com/doc
    - https://training.github.com
- Denx U-Boot Manual
  - Complete documentation from the folks who wrote Das U-Boot
    - http://www.denx.de/wiki/U-Boot/Documentation
- Free-Electrons:
  - Complete training materials posted free
    - http://free-electrons.com/docs/
- Device Tree for Dummies
  - http://events.linuxfoundation.org/sites/events/files/slides/petazzoni-device-tree-dummies.pdf

The Two Best Sources for Linux Development Information

- Linux Kernel Documentation
  - The most complete and most essential Linux kernel documentation
  - Included with the Linux kernel source code
    - <local GIT repo>/Documentation


Documentation : bash – Konsole
File Edit View Bookmarks Settings Help

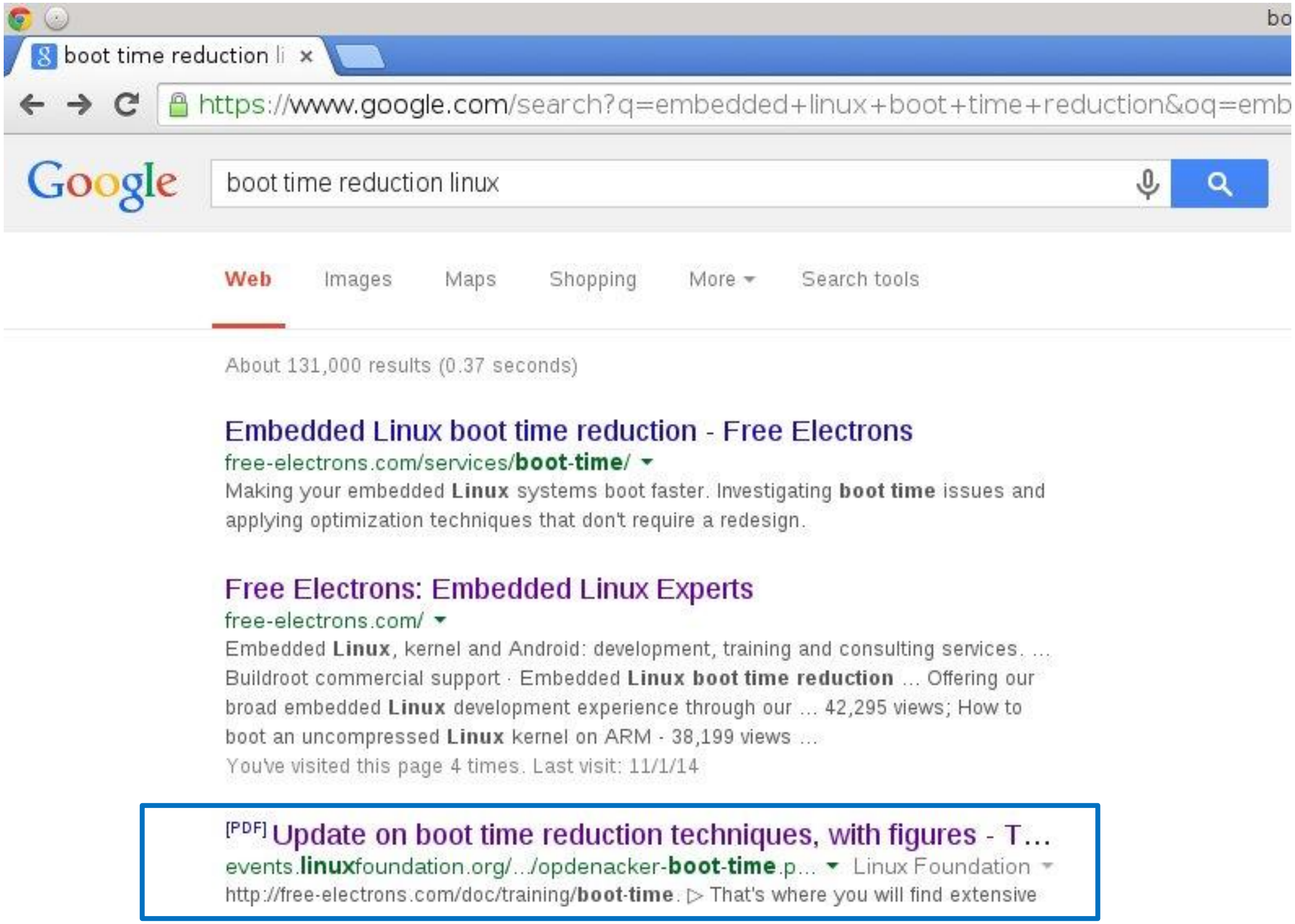
```
/data/linux_repos/linux-socfpga/Documentation>>grep -ir tickless *
timers/highres.txt:The implementation leaves room for further development like full tickless
timers/NO_HZ.txt:tickless".  The remainder of this document will use "dyntick-idle mode".
/data/linux_repos/linux-socfpga/Documentation>
```



```
		NO_HZ: Reducing Scheduling-Clock Ticks


This document describes Kconfig options and boot parameters that can
reduce the number of scheduling-clock interrupts, thereby improving energy
efficiency and reducing OS jitter.  Reducing OS jitter is important for
some types of computationally intensive high-performance computing (HPC)
applications and for real-time applications.

There are three main ways of managing scheduling-clock interrupts
(also known as "scheduling-clock ticks" or simply "ticks"):

1.	Never omit scheduling-clock ticks (CONFIG_HZ_PERIODIC=y or
	CONFIG_NO_HZ=n for older kernels).  You normally will -not-
	want to choose this option.
```

The Two Best Sources for Linux Development Information

- An open source OS breeds open source information



boot time reduction li ×

https://www.google.com/search?q=embedded+linux+boot+time+reduction&oq=emb

boot time reduction linux

Web Images Maps Shopping More Search tools

About 131,000 results (0.37 seconds)

Embedded Linux boot time reduction - Free Electrons
free-electrons.com/services/**boot-time**/
Making your embedded **Linux** systems boot faster. Investigating **boot time** issues and applying optimization techniques that don't require a redesign.

Free Electrons: Embedded Linux Experts
free-electrons.com/
Embedded **Linux**, kernel and Android: development, training and consulting services. ... Buildroot commercial support · Embedded **Linux boot time reduction** ... Offering our broad embedded **Linux** development experience through our ... 42,295 views; How to boot an uncompressed **Linux** kernel on ARM - 38,199 views ...
You've visited this page 4 times. Last visit: 11/1/14

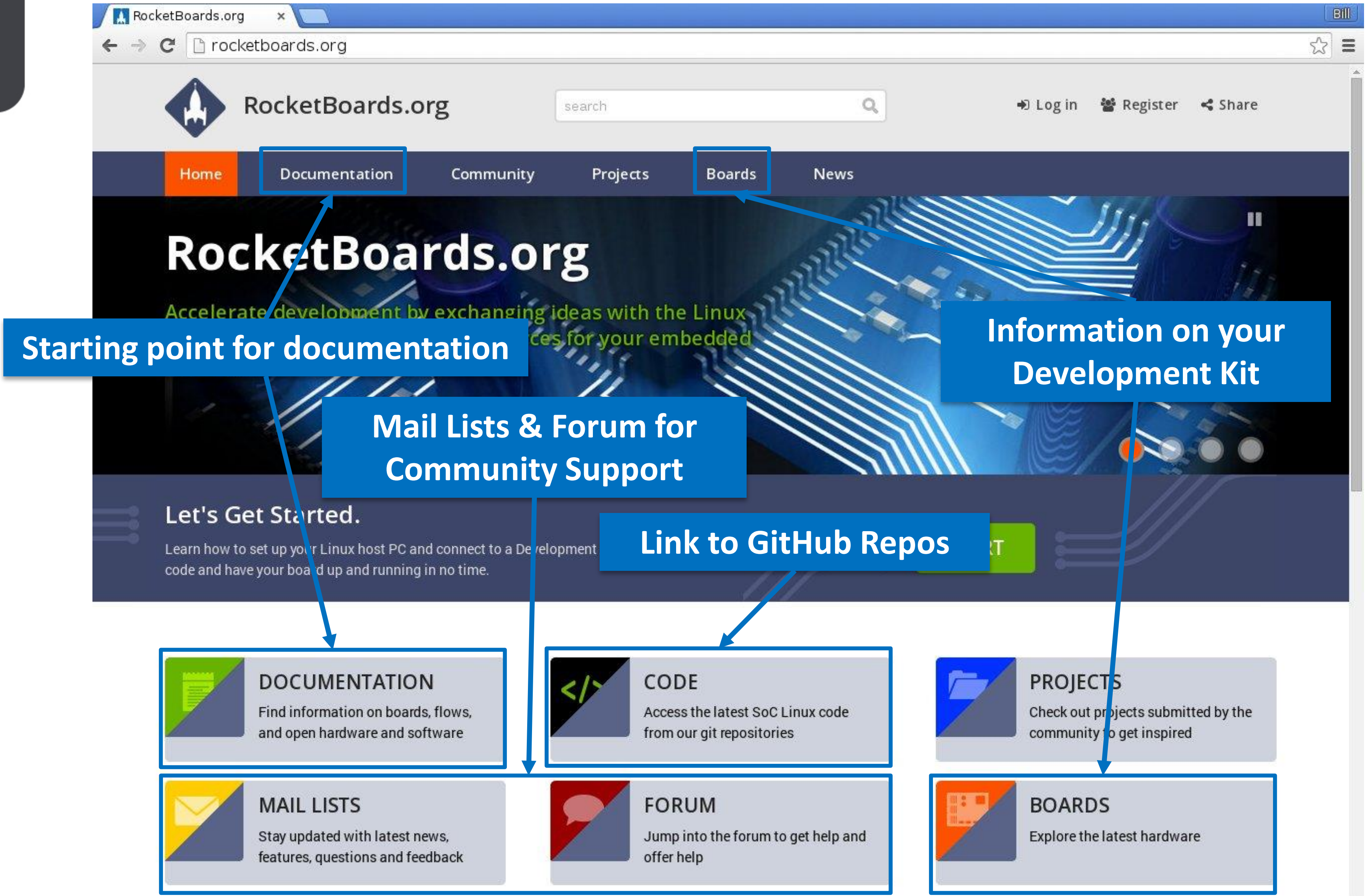
[PDF] **Update on boot time reduction techniques, with figures - T...**
events.**linux**foundation.org/.../opdenacker-**boot-time**.p... Linux Foundation
http://free-electrons.com/doc/training/**boot-time**. ▷ That's where you will find extensive

RocketBoards.org – Altera SoC Linux Community Portal

- The source for SoC FPGA Linux info
  - Golden System Reference Design (GSRD)
  - Updates on latest releases
  - Step-by-step getting started guides
- SoC FPGA Mailing List - RFI
  - Active community participation in answering SoC FPGA and Linux questions
- Example Projects, Applications, and Designs
  - From Altera and the SoC community
- Enables the SoC community to support Linux

RocketBoards.org

RocketBoards.org Resources



RocketBoards.org

Accelerate development by exchanging ideas with the Linux ... ces for your embedded

Let's Get Started.

Learn how to set up your Linux host PC and connect to a Development ... code and have your board up and running in no time.

DOCUMENTATION
Find information on boards, flows, and open hardware and software

CODE
Access the latest SoC Linux code from our git repositories

PROJECTS
Check out projects submitted by the community to get inspired

MAIL LISTS
Stay updated with latest news, features, questions and feedback

FORUM
Jump into the forum to get help and offer help

BOARDS
Explore the latest hardware

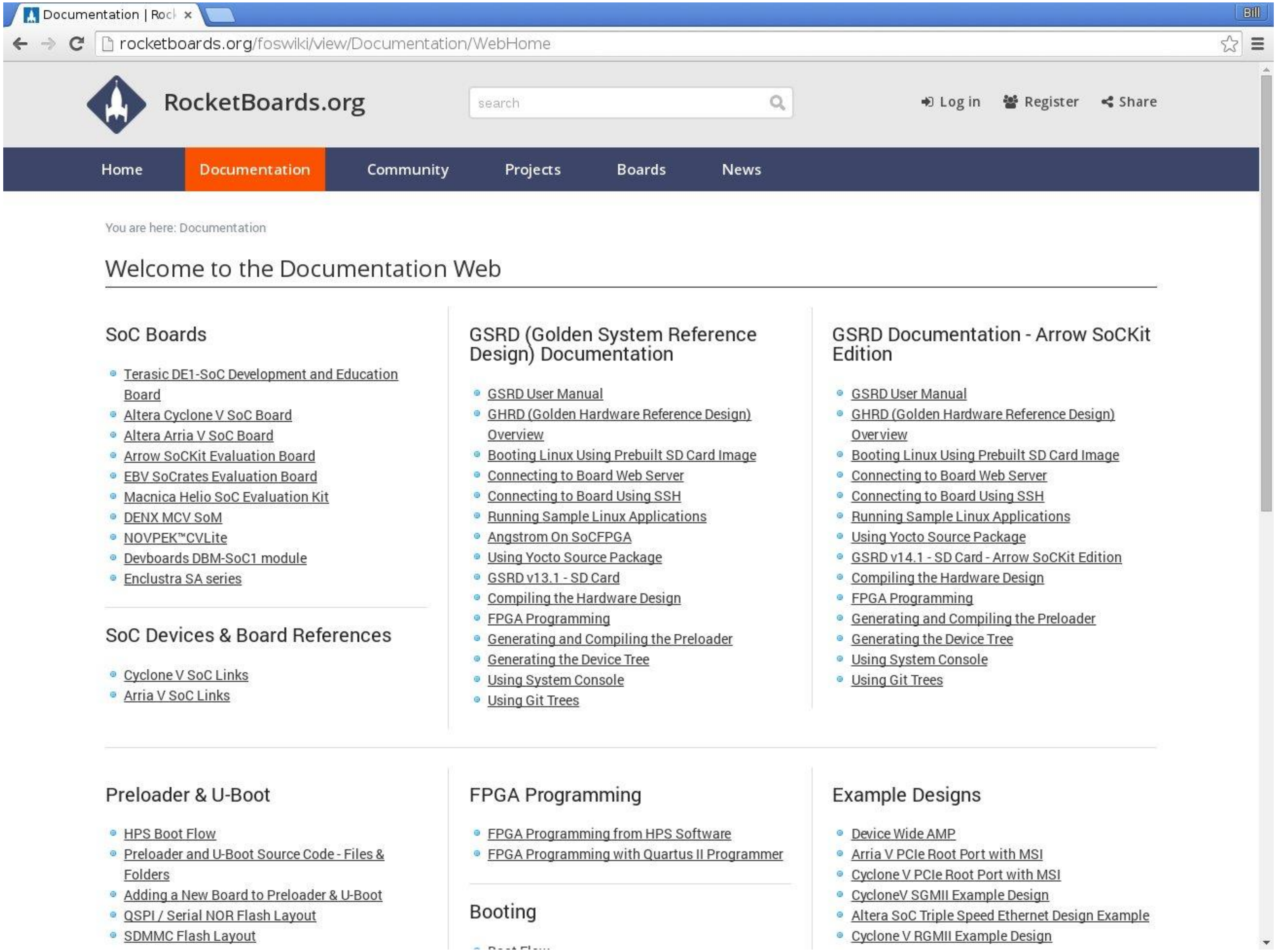
Starting point for documentation

Information on your Development Kit

Mail Lists & Forum for Community Support

Link to GitHub Repos

RocketBoards.org Documentation



You are here: Documentation

Welcome to the Documentation Web

SoC Boards

- Terasic DE1-SoC Development and Education Board
- Altera Cyclone V SoC Board
- Altera Arria V SoC Board
- Arrow SoCKit Evaluation Board
- EBV SoCrates Evaluation Board
- Macnica Helio SoC Evaluation Kit
- DENX MCV SoM
- NOVPEK™CVLite
- Devboards DBM-SoC1 module
- Enclustra SA series

SoC Devices & Board References

- Cyclone V SoC Links
- Arria V SoC Links

GSRD (Golden System Reference Design) Documentation

- GSRD User Manual
- GHRD (Golden Hardware Reference Design) Overview
- Booting Linux Using Prebuilt SD Card Image
- Connecting to Board Web Server
- Connecting to Board Using SSH
- Running Sample Linux Applications
- Angstrom On SoCFPGA
- Using Yocto Source Package
- GSRD v13.1 - SD Card
- Compiling the Hardware Design
- FPGA Programming
- Generating and Compiling the Preloader
- Generating the Device Tree
- Using System Console
- Using Git Trees

GSRD Documentation - Arrow SoCKit Edition

- GSRD User Manual
- GHRD (Golden Hardware Reference Design) Overview
- Booting Linux Using Prebuilt SD Card Image
- Connecting to Board Web Server
- Connecting to Board Using SSH
- Running Sample Linux Applications
- Using Yocto Source Package
- GSRD v14.1 - SD Card - Arrow SoCKit Edition
- Compiling the Hardware Design
- FPGA Programming
- Generating and Compiling the Preloader
- Generating the Device Tree
- Using System Console
- Using Git Trees

Preloader & U-Boot

- HPS Boot Flow
- Preloader and U-Boot Source Code - Files & Folders
- Adding a New Board to Preloader & U-Boot
- QSPI / Serial NOR Flash Layout
- SDMMC Flash Layout

FPGA Programming

- FPGA Programming from HPS Software
- FPGA Programming with Quartus II Programmer

Booting

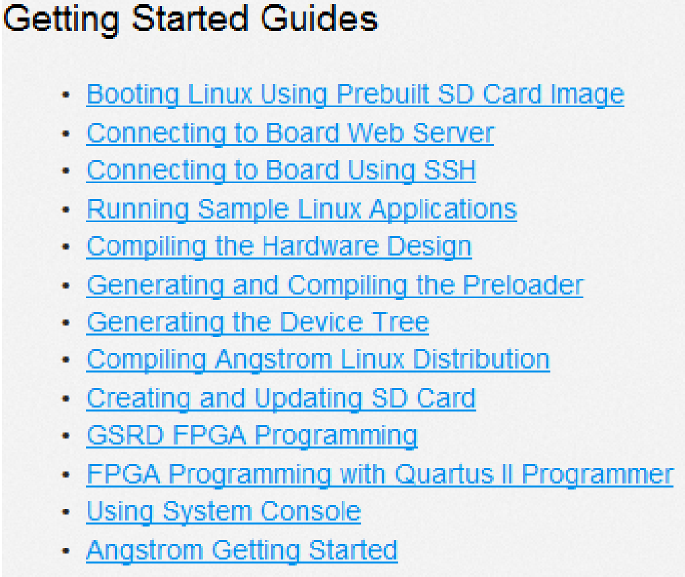
- Boot Flow

Example Designs

- Device Wide AMP
- Arria V PCIe Root Port with MSI
- Cyclone V PCIe Root Port with MSI
- CycloneV SGMII Example Design
- Altera SoC Triple Speed Ethernet Design Example
- Cyclone V RGMII Example Design

RocketBoards.org – Useful Links

- GSRD User Manual
  - The best starting point for Linux development
  - http://www.rocketboards.org/foswiki/Documentation/GSRD



Getting Started Guides

- Booting Linux Using Prebuilt SD Card Image
- Connecting to Board Web Server
- Connecting to Board Using SSH
- Running Sample Linux Applications
- Compiling the Hardware Design
- Generating and Compiling the Preloader
- Generating the Device Tree
- Compiling Angstrom Linux Distribution
- Creating and Updating SD Card
- GSRD FPGA Programming
- FPGA Programming with Quartus II Programmer
- Using System Console
- Angstrom Getting Started

- Device Tree Generator User Guide
  - http://www.rocketboards.org/foswiki/Documentation/GSRD141DeviceTreeGenerator
- Programming FPGA from HPS
  - http://www.rocketboards.org/foswiki/Documentation/GSRD131ProgrammingFPGA
- GSRD Releases
  - http://releases.rocketboards.org

Several Ways to Learn!

- **Instructor-led training**
  - Face to face with an Altera expert Training Engineer
  - 20+ courses to choose from (8 hour classes)

- **Virtual classes (taught via WebEX)**
  - Can ask questions to Altera expert Training Engineer
  - Course content same as instructor-led classes
    (1/2 day sessions)

- **Online training (free and always available)**
  - 200+ topics available (~30 minutes in length)

- **Videos (free and always available)**
  - YouTube videos (~4 minutes each)

SoC Classes Available

- Instructor-led or virtual classes
  - Designing with an ARM-based SoC
  - Developing Software for an ARM-based SoC
- Online classes
  - Hardware Design Flow for an ARM-based SoC
  - Software Design Flow for an ARM-based SoC
  - SoC Hardware Overview: the Microprocessor Unit
  - SoC Hardware Overview: Interconnect and Memory
  - SoC Hardware Overview: System Management, Debug, and General Purpose Peripherals
  - SoC Hardware Overview: Flash Controllers and Interface Protocols
  - SoC Bare-metal Programming and Hardware Libraries
  - Getting Started with Linux for Altera SoCs

Essential SoC Software Tools Online Videos

- ARM DS-5 Altera Edition Toolchain
  - https://youtu.be/HV6NHr6gLx0
- DS-5 Altera Edition: Bare-metal Debug and Trace
  - https://youtu.be/u_xKybPhcHI
- DS-5 Altera Edition: FPGA-adaptive Linux Kernel Debug and Trace
  - https://youtu.be/lrR-SfVZd18
- Debugging Linux applications on the Altera SoC with ARM DS-5
  - https://youtu.be/ZcGQEjkYWOc
- FPGA-adaptive debug on the Altera SoC using ARM DS-5
  - https://youtu.be/2NBcUv2Txbl
- Streamline Profiling on Altera SoC FPGA. Part 1 - Setup
  - https://youtu.be/X-k9lmXQTio
- Streamline Profiling on Altera SoC FPGA. Part 2 - Running Streamline
  - https://youtu.be/Tzbd7qldKqY

Essential SoC Hardware Documentation Resources

- Hard Processor System Technical Reference Manuals
  - Available in Device Handbooks:
    - https://www.altera.com/products/soc/portfolio/cyclone-v-soc/support.html
    - https://www.altera.com/products/soc/portfolio/arria-v-soc/support.html
    - https://www.altera.com/products/soc/portfolio/arria-10-soc/support.html
  - Contain Functional Descriptions Peripheral
  - Contain Control Register Address Map and Definitions
    - These are also available online at the links above in HTML and PDF formats
- HPS SoC Boot Guide
  - Cyclone V SoC & Arria V SoC: AN709 - HPS SoC Boot Guide
  - Arria 10 SoC: included in HPS TRM in Arria 10 Device Handbook
- ARM Documentation Site
  - Documentation available for all ARM IP
    - Cortex-A9 & A53 MP Cores, FPU, NEON, GIC, ARM Peripherals, etc.
  - Requires free registration
  - Refer to HPS TRM for IP core names and revision information
  - http://infocenter.arm.com/help/index.jsp

Essential SoC Software Documentation Resources

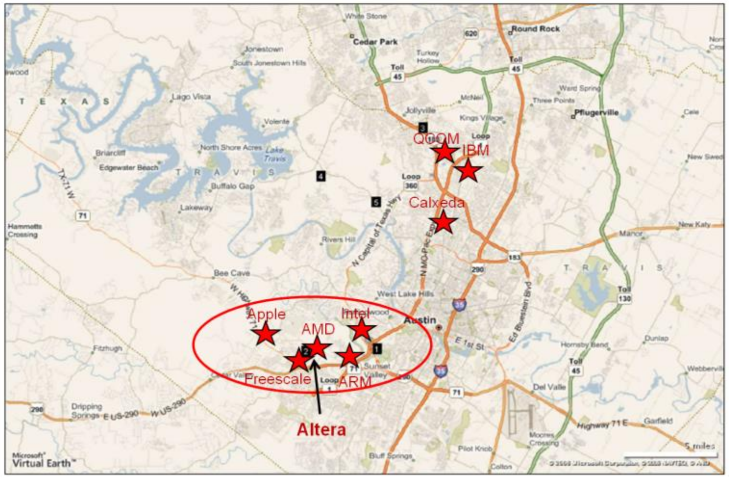
- Altera SoC Embedded Design Software (SoC EDS) Tools
  - User Guide:
    https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_soc_eds.pdf
    - Linux & Baremetal Software Development Tools Overview
    - HPS Preloader User Guide
    - HPS Flash Programmer User Guide
    - SD Card Boot Utility
  - Getting Started Guides: Preloader, Linux, Bare Metal, Debug, HW Library
    http://www.alterawiki.com/wiki/SoCEDSGettingStarted
  - SoC HPS Release Notes
  - SoC Abstraction Layer (SoCAL) API Reference
    - <SoC EDS install dir>/ip/altera/hps/altera_hps/doc/socal/html/index.html
  - Hardware Manager API Reference
    - <SoC EDS install dir>/ip/altera/hps/altera_hps/doc/hwmgr/html/index.html
  - GCC Documentation
    - <SoC EDS install dir>/ds-5/documents/gcc/getting_started.html
  - Bare Metal Compiler
    - <SoC EDS installation directory>/host_tools/mentor/gnu/arm/baremetal/share/doc/sourceryg++-arm-altera-eabi

SoC Device Overview

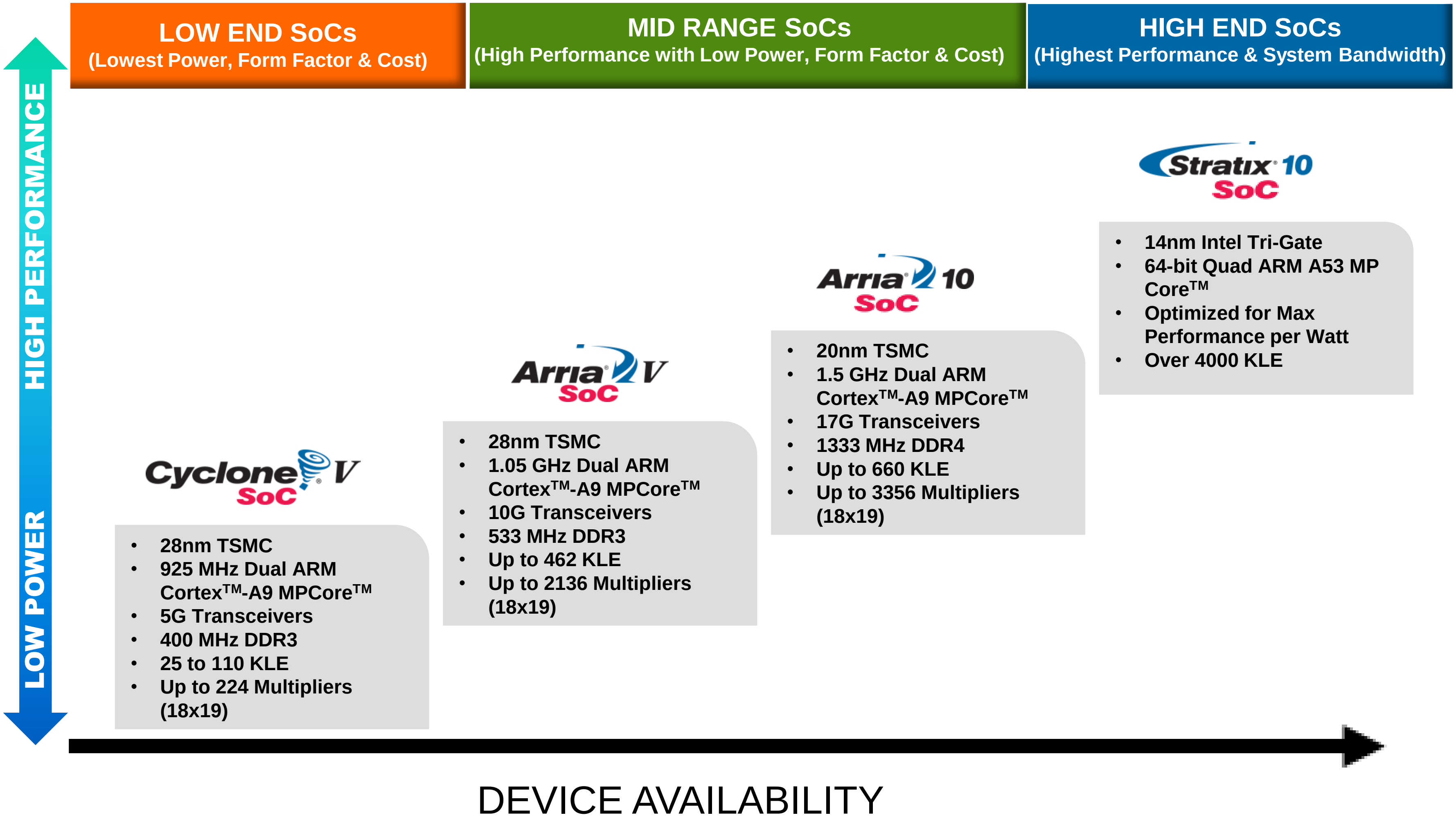
Altera Investment in Embedded Technologies

- Altera established Austin Technology Center (ATC) in 2011

- Altera's primary embedded engineering center
- Austin provides access to one of the richest embedded processing talent bases in the world



Altera SoC Product Portfolio



LOW END SoCs
(Lowest Power, Form Factor & Cost)

MID RANGE SoCs
(High Performance with Low Power, Form Factor & Cost)

HIGH END SoCs
(Highest Performance & System Bandwidth)

HIGH PERFORMANCE

LOW POWER

Cyclone V SoC

- 28nm TSMC
- 925 MHz Dual ARM Cortex^{TM}-A9 MPCore^{TM}
- 5G Transceivers
- 400 MHz DDR3
- 25 to 110 KLE
- Up to 224 Multipliers (18x19)

Arria V SoC

- 28nm TSMC
- 1.05 GHz Dual ARM Cortex^{TM}-A9 MPCore^{TM}
- 10G Transceivers
- 533 MHz DDR3
- Up to 462 KLE
- Up to 2136 Multipliers (18x19)

Arria 10 SoC

- 20nm TSMC
- 1.5 GHz Dual ARM Cortex^{TM}-A9 MPCore^{TM}
- 17G Transceivers
- 1333 MHz DDR4
- Up to 660 KLE
- Up to 3356 Multipliers (18x19)

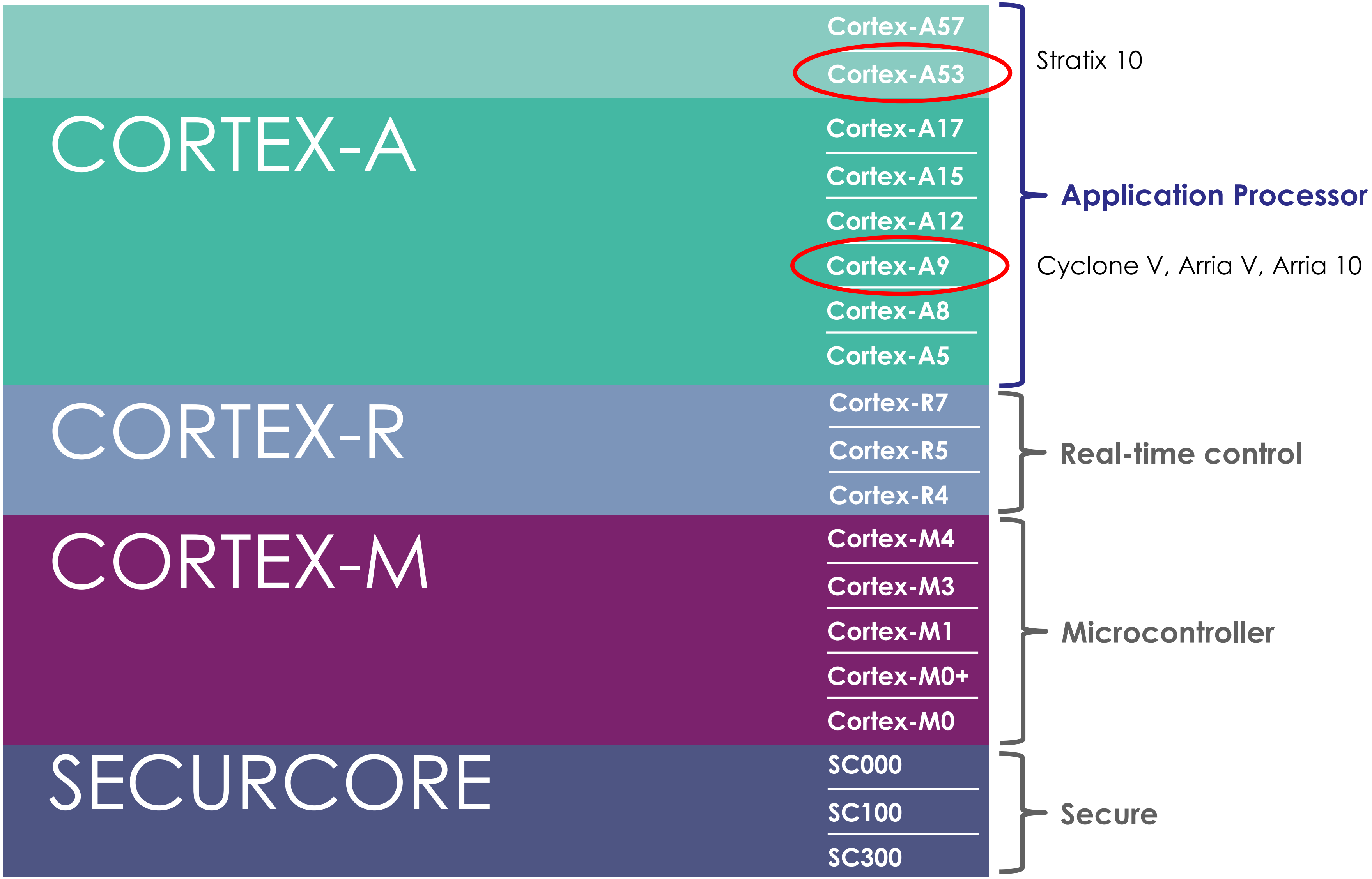
Stratix 10 SoC

- 14nm Intel Tri-Gate
- 64-bit Quad ARM A53 MP Core^{TM}
- Optimized for Max Performance per Watt
- Over 4000 KLE

DEVICE AVAILABILITY

SoC devices available across entire product portfolio …

ARM Public Processor Offering



CORTEX-A

Cortex-A57

Cortex-A53

Cortex-A17

Cortex-A15

Cortex-A12

Cortex-A9

Cortex-A8

Cortex-A5

Stratix 10

Application Processor

Cyclone V, Arria V, Arria 10

CORTEX-R

Cortex-R7

Cortex-R5

Cortex-R4

Real-time control

CORTEX-M

Cortex-M4

Cortex-M3

Cortex-M1

Cortex-M0+

Cortex-M0

Microcontroller

SECURCORE

SC000

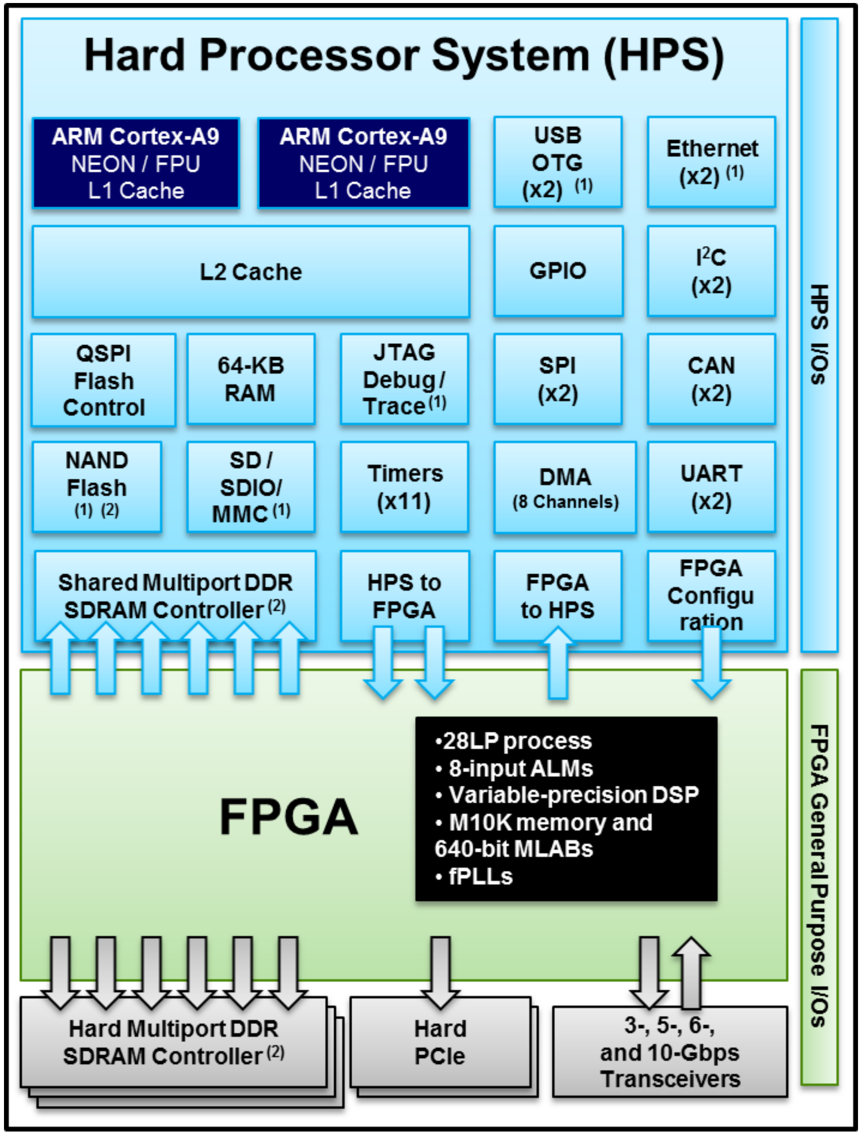
SC100

SC300

Secure

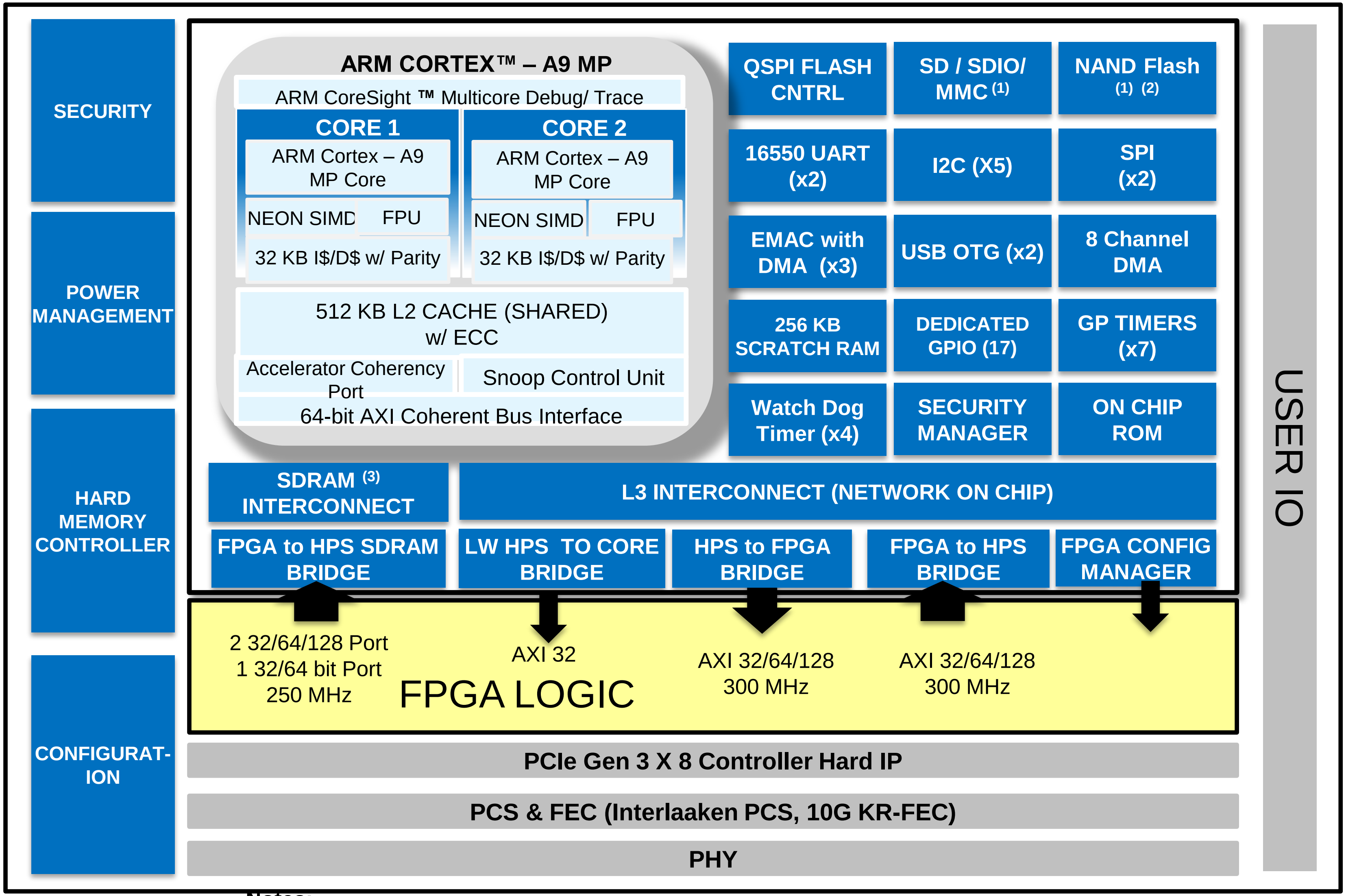
28nm SoC System Architecture

- Processor
  - Dual-core ARM® Cortex™-A9 MPCore™ processor
  - Up to 5,250 MIPS (1050 MHz per core maximum)
  - NEON coprocessor with double-precision FPU
  - 32-KB/32-KB L1 caches per core
  - 512-KB shared L2 cache
- Multiport SDRAM controller
  - DDR3, DDR3L, DDR2, LPDDR2
  - Integrated ECC support
- High-bandwidth on-chip interfaces
  - > 125-Gbps HPS-to-FPGA interface
  - > 125-Gbps FPGA-to-SDRAM interface
- Cost- and power-optimized FPGA fabric
  - Lowest power transceivers
  - Up to 1,600 GMACS, 300 GFLOPS
  - Up to 25Mb on-chip RAM
  - More hard intellectual property (IP): PCIe® and memory controllers



Notes:
(1) Integrated direct memory access (DMA)
(2) Integrated ECC

Arria 10 HPS Block Diagram



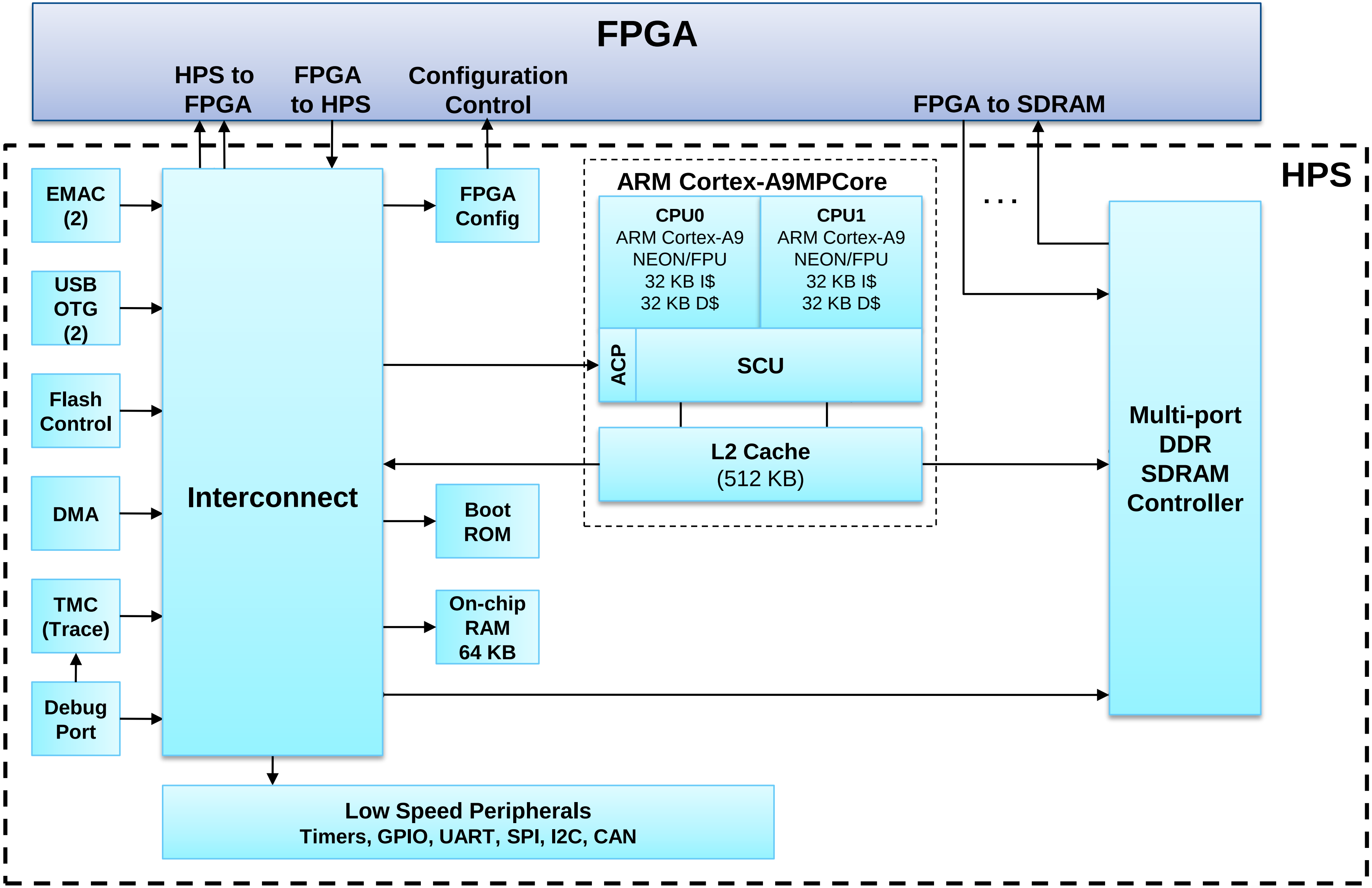
Notes:

(1) Integrated direct memory access (DMA)

(2) Integrated ECC

(3) DDR3/4 & LP DDR3 SDRAM Support fo HPS Memory

High-Level Block Diagram



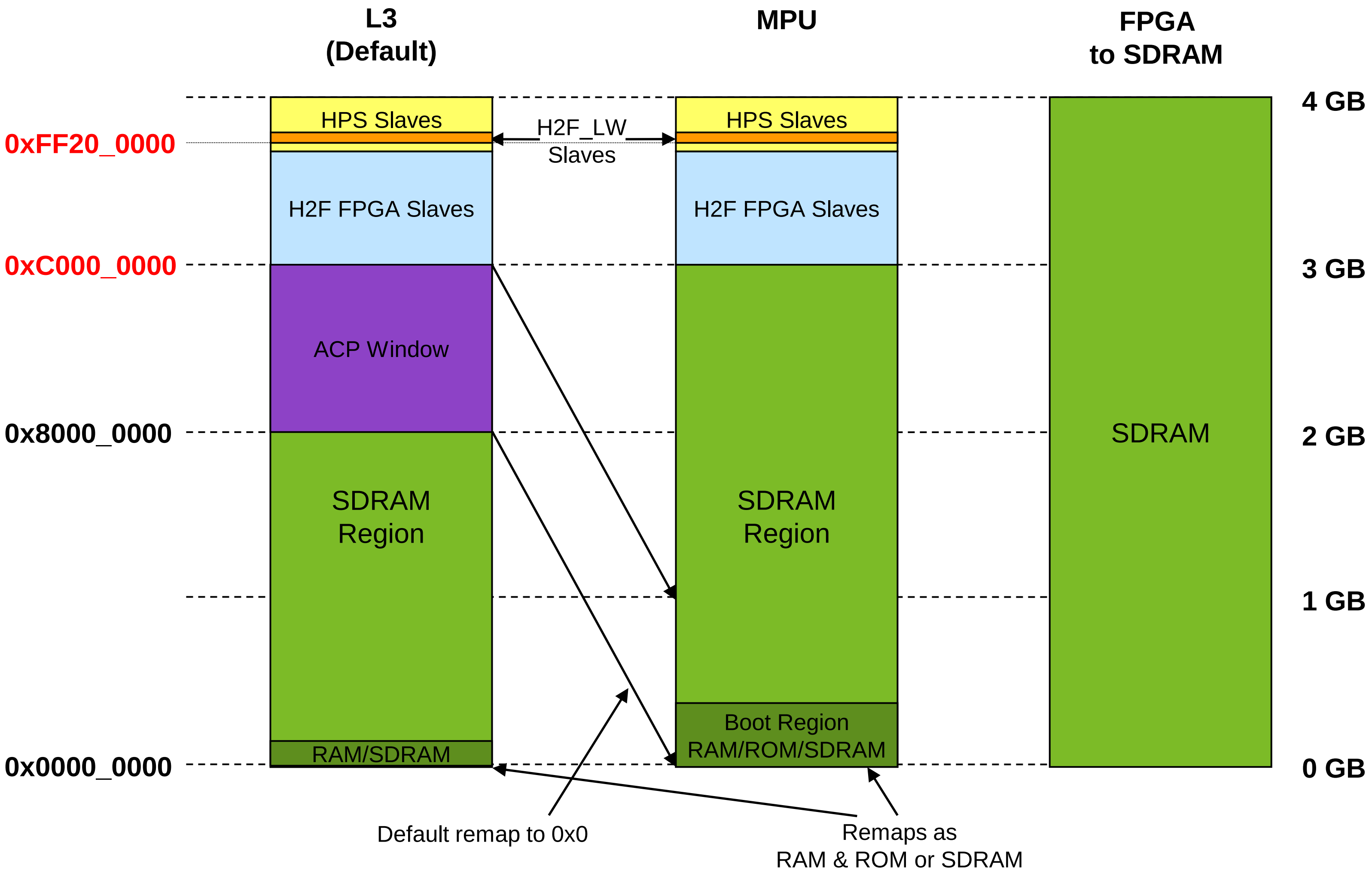
A Comparison: Cyclone V SoC, Arria V SoC, Arria 10 SoC

Metric	Cyclone V SoC	Arria V SoC	Arria 10 SoC
Technology	28nm	28nm	20nm
Processor Performance	925 MHz	1.05 GHz	1.5 GHz
Total Power Dissipation	100%	100%	60% (40% Lower)
Max PCI Express Hard IP	Gen 2 x4	Gen 2 x8	Gen 3 x8
Memory Devices Supported	DDR2, DDR3, DDR3L, LPDDR2	DDR2, DDR3, DDR3L, LPDDR2	DDR4/3, LPDDR2/3, QDRIV, RLDRAM III, Hybrid Memory Cube
Max. HPS DDR Data-Width	40-bit (32-bit + ECC)	40-bit (32-bit + ECC)	72-bit (64-bit + ECC)
EMAC Cores	EMAC x 2	EMAC x 2	EMAC x 3
NAND Device Supported	8-bit	8-bit	8-bit and 16-bit
SD/MMC devices supported	SD/SDIO/MMC	SD/SDIO/MMC	SD/SDIO/MMC 4.5 with eMMC
FPGA Logic Density Range (LEs)	25 - 110K	370 - 450K	160 - 660K
FPGA Core Performance	260 MHz	307 MHz	500 MHz

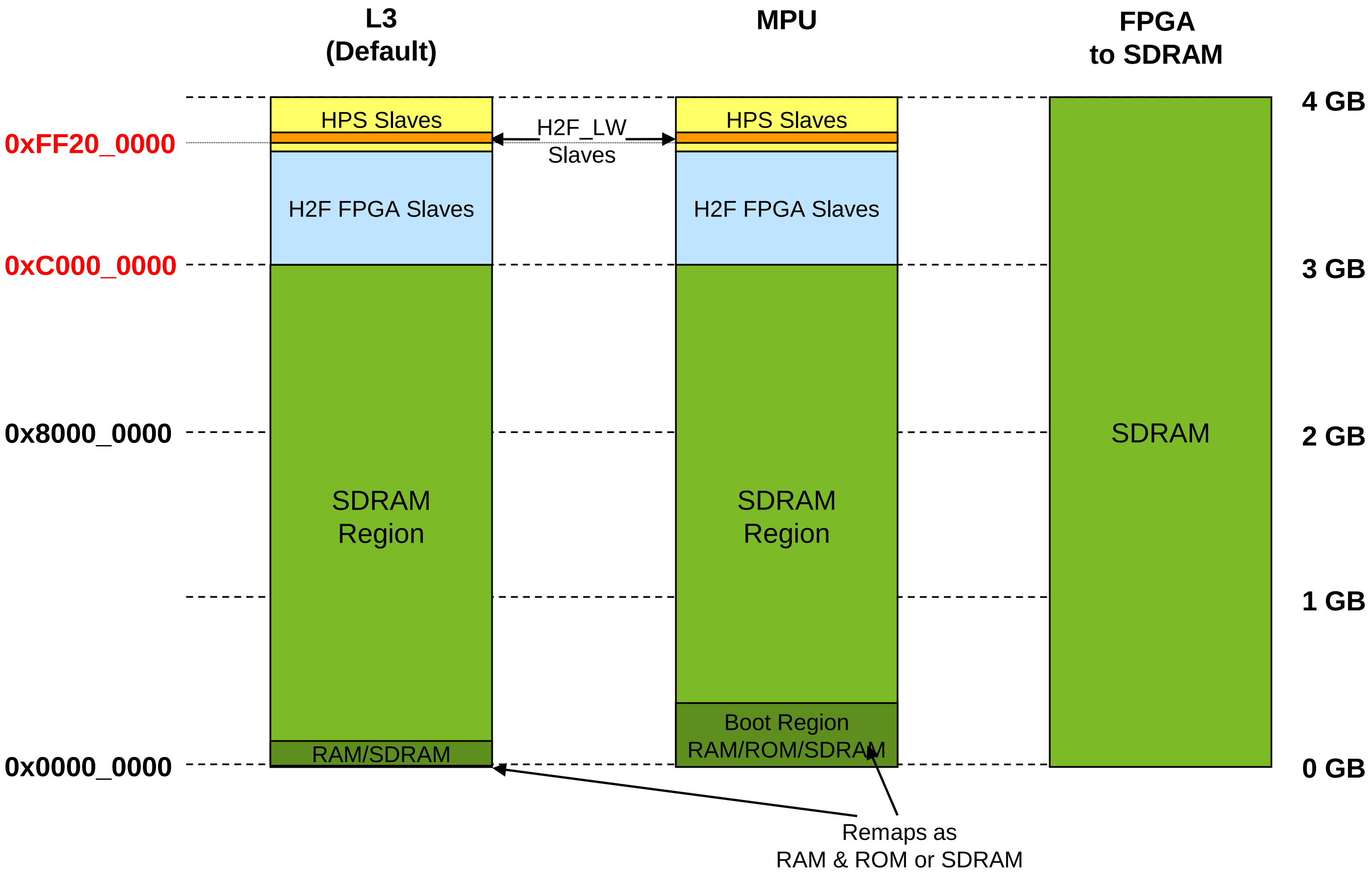
SoC Physical Address Map

Essential HW information for SW Developers

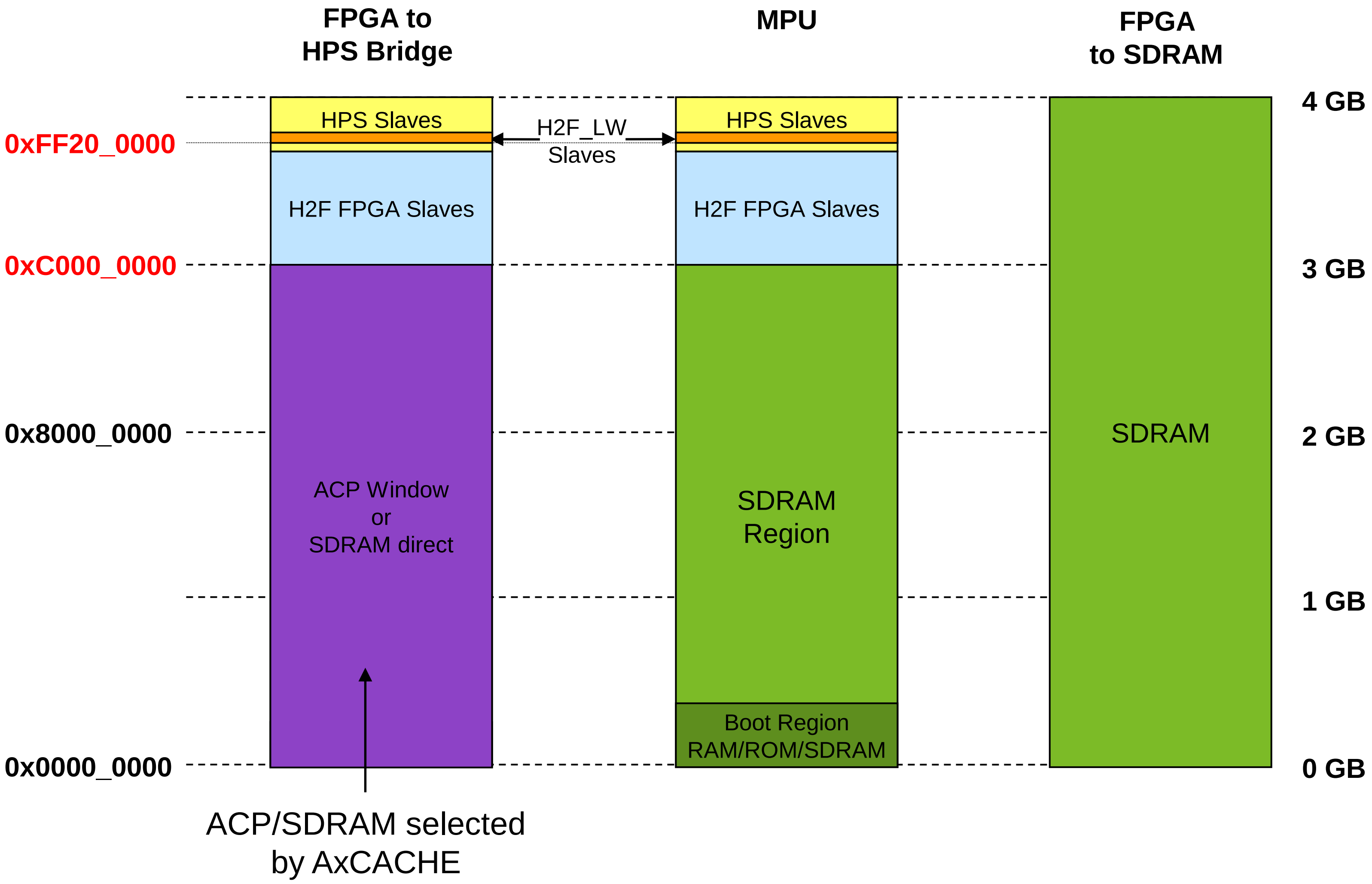
Cyclone V & Arria V SoC HPS Physical Memory Map



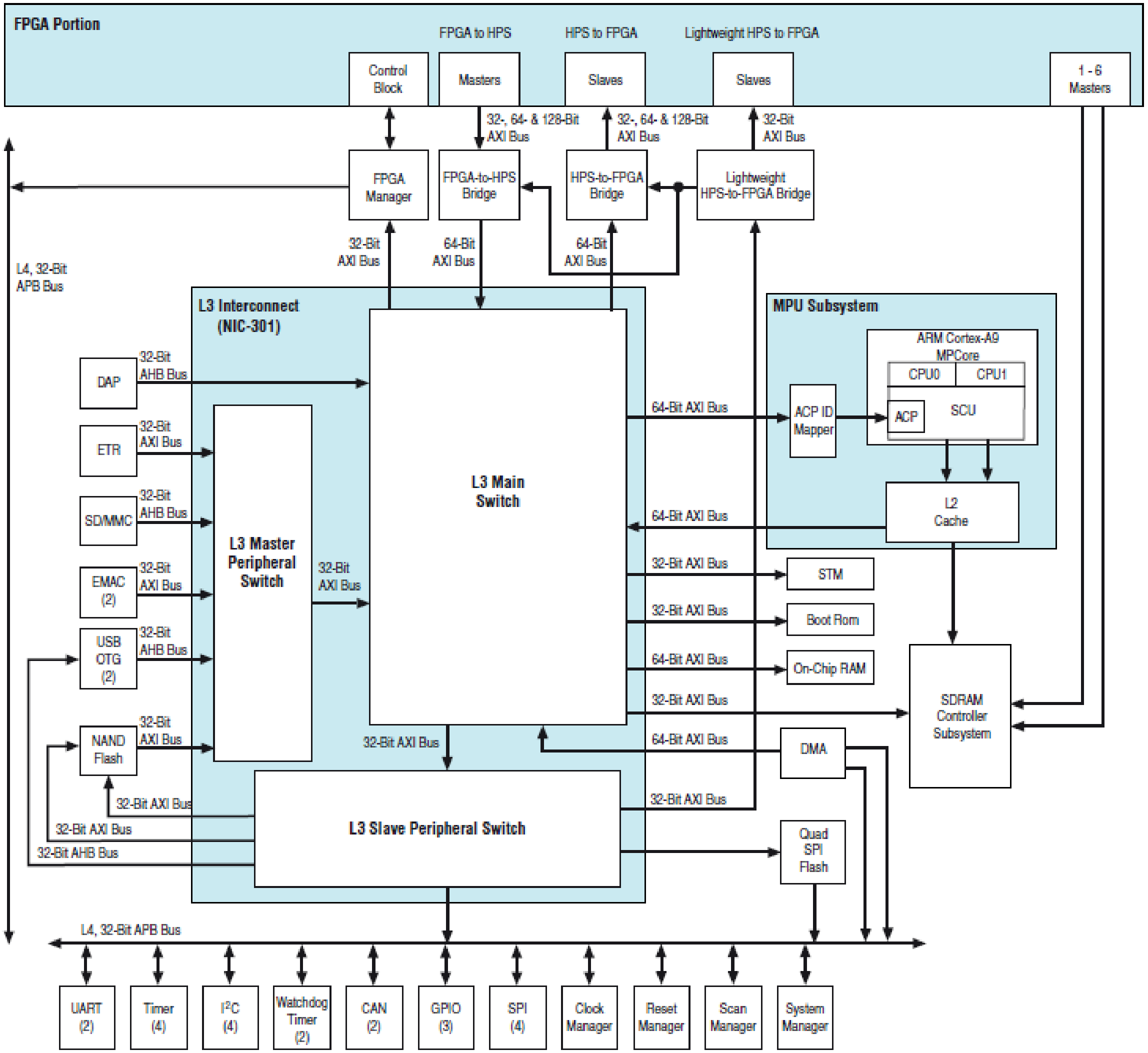
Arria 10 SoC HPS Physical Memory Map



Arria 10 SoC HPS Physical Memory Map

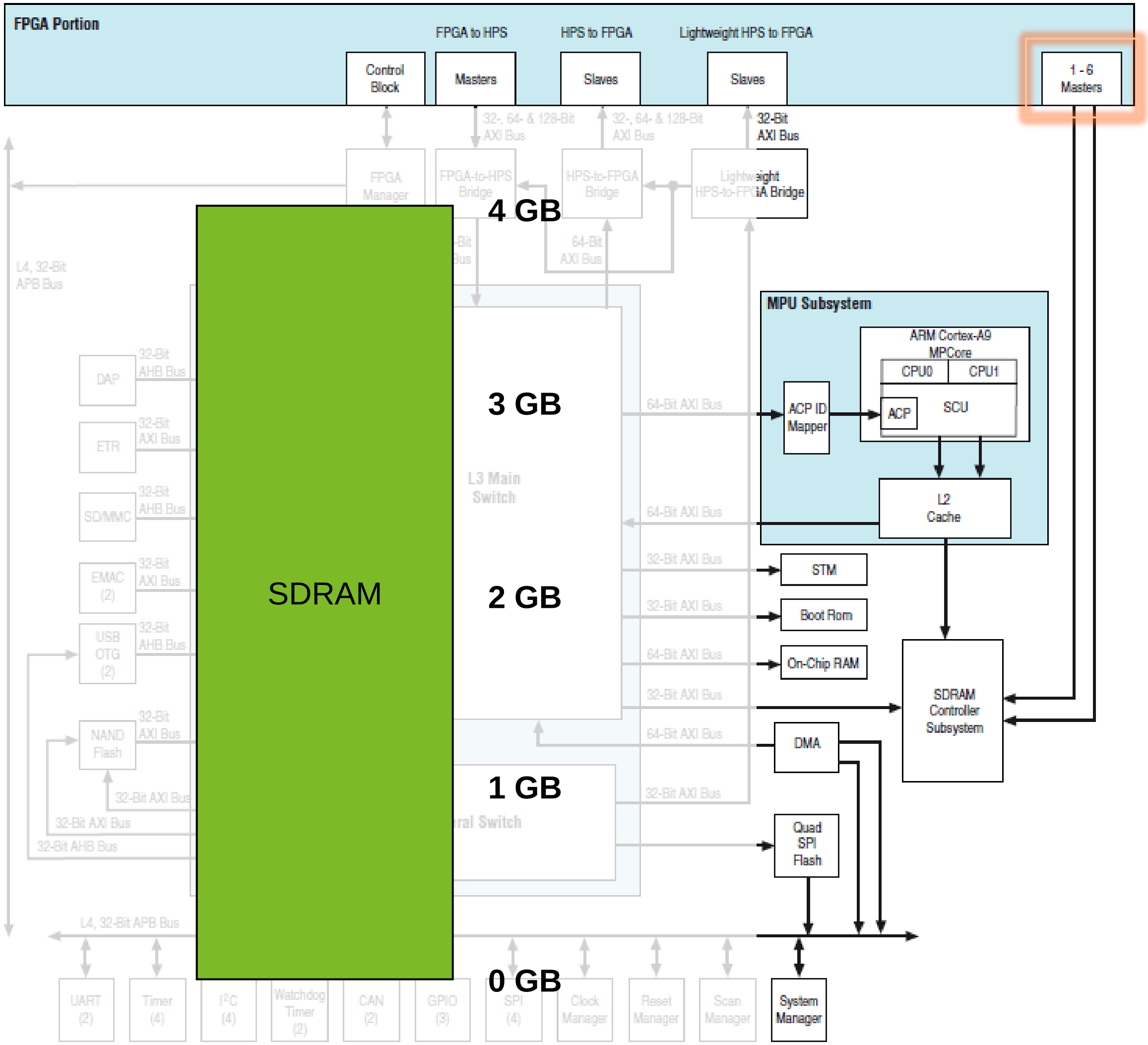


Physical Address Mapping



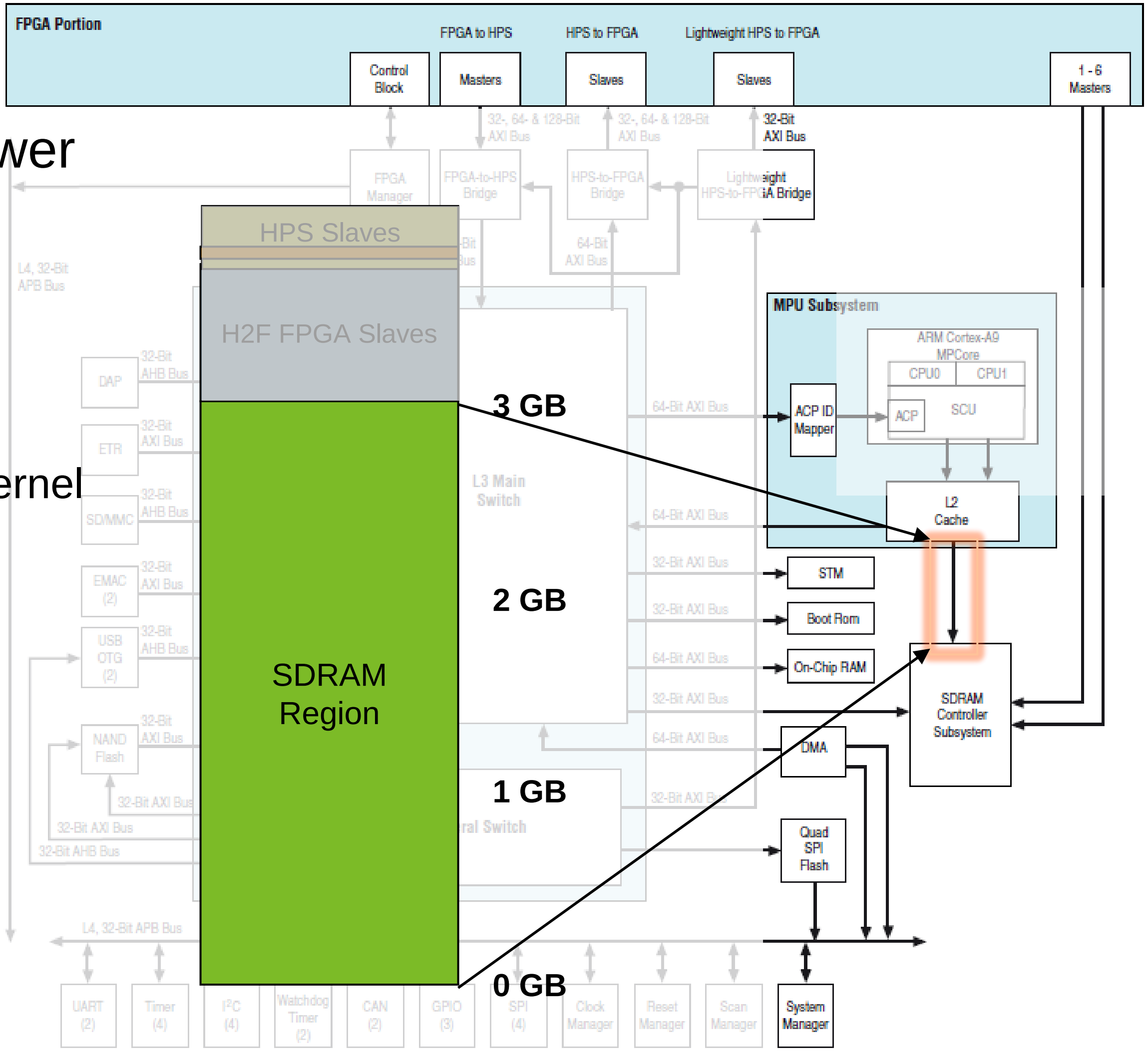
Physical Address Mapping – FPGA to SDRAM

- FPGA Masters have access to full 4GB of SDRAM address space
  - Subject to MPFE MPU restrictions
  - No coherency
  - No virtual addressing



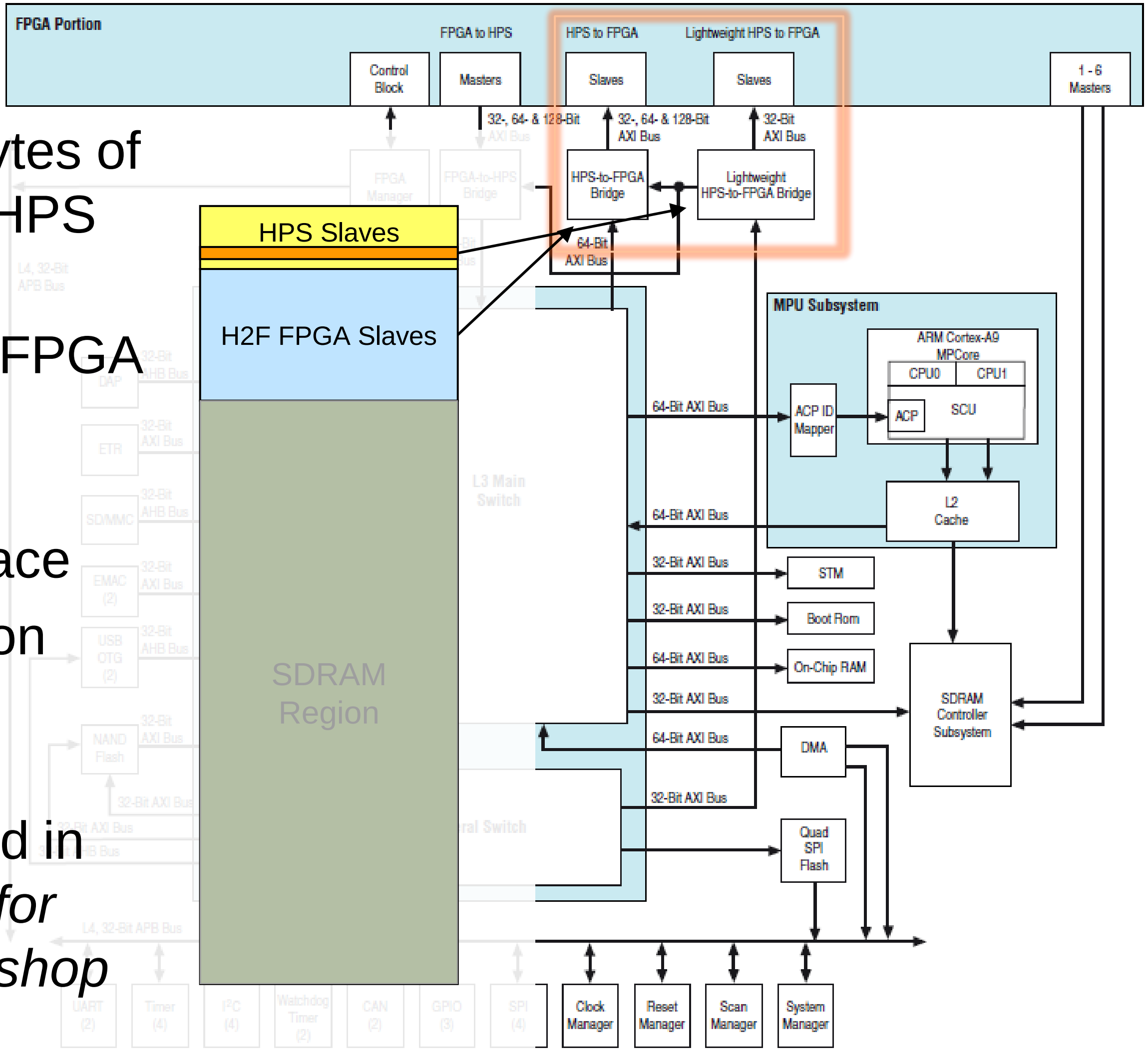
Physical Address Mapping – MPU

- MPU has access to the lower 3 GBytes of SDRAM
- Kernel manages and can allocate memory in this 3GByte space
  - Allocate for both user and kernel space
  - Allocatable on 4K Byte Boundaries (page size)



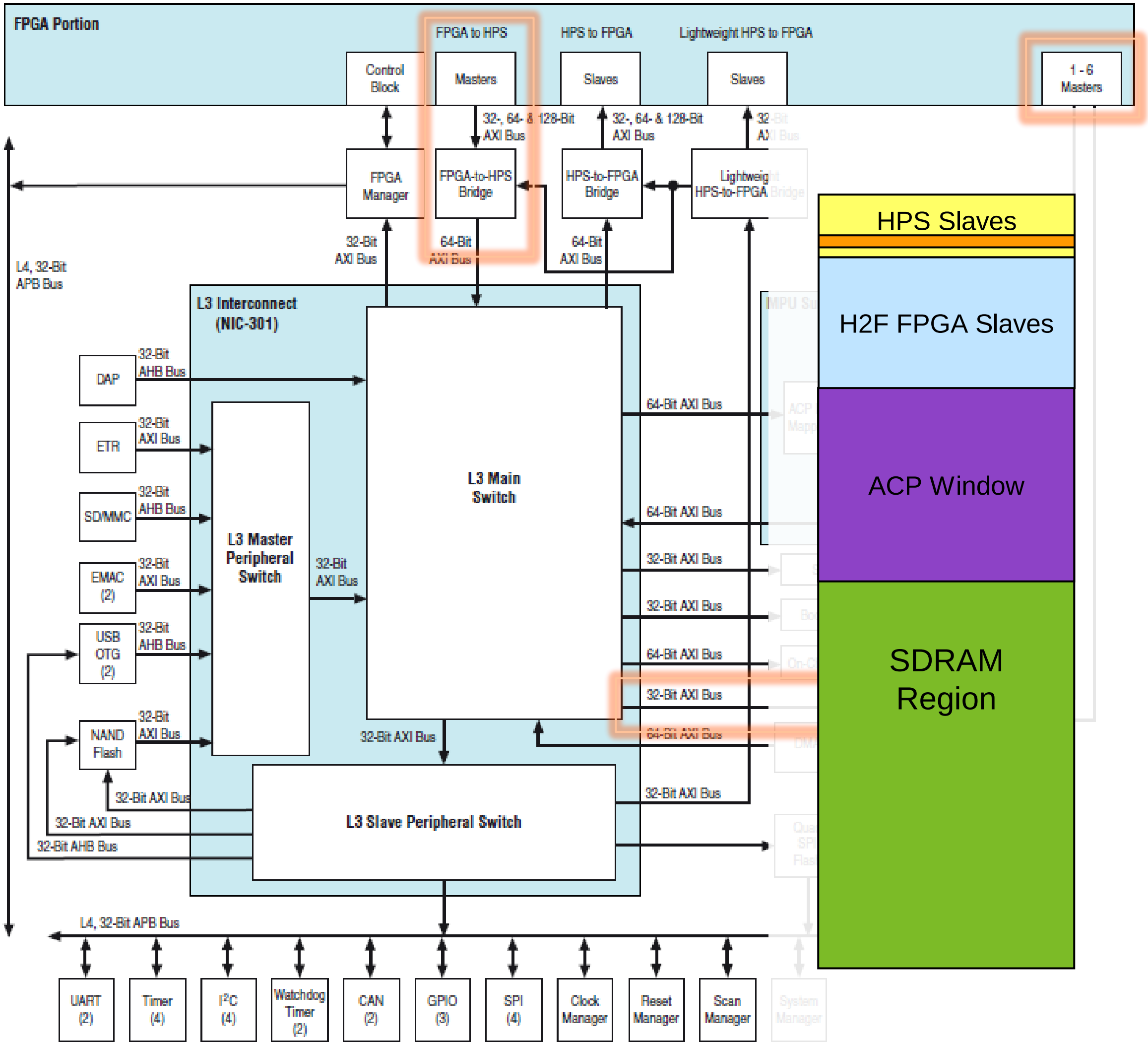
Physical Address Mapping – MPU to FPGA

- MPU can access 960 MBytes of FPGA address space via HPS to FPGA Bridge
- MPU can access 2 MB of FPGA address space via HPS to FPGA Lightweight bridge
- Not allocatable in user space
- Space FPGA peripherals on Linux page size (4KB) boundaries
- Access methods discussed in *Developing Linux Drivers for Custom Peripherals Workshop*

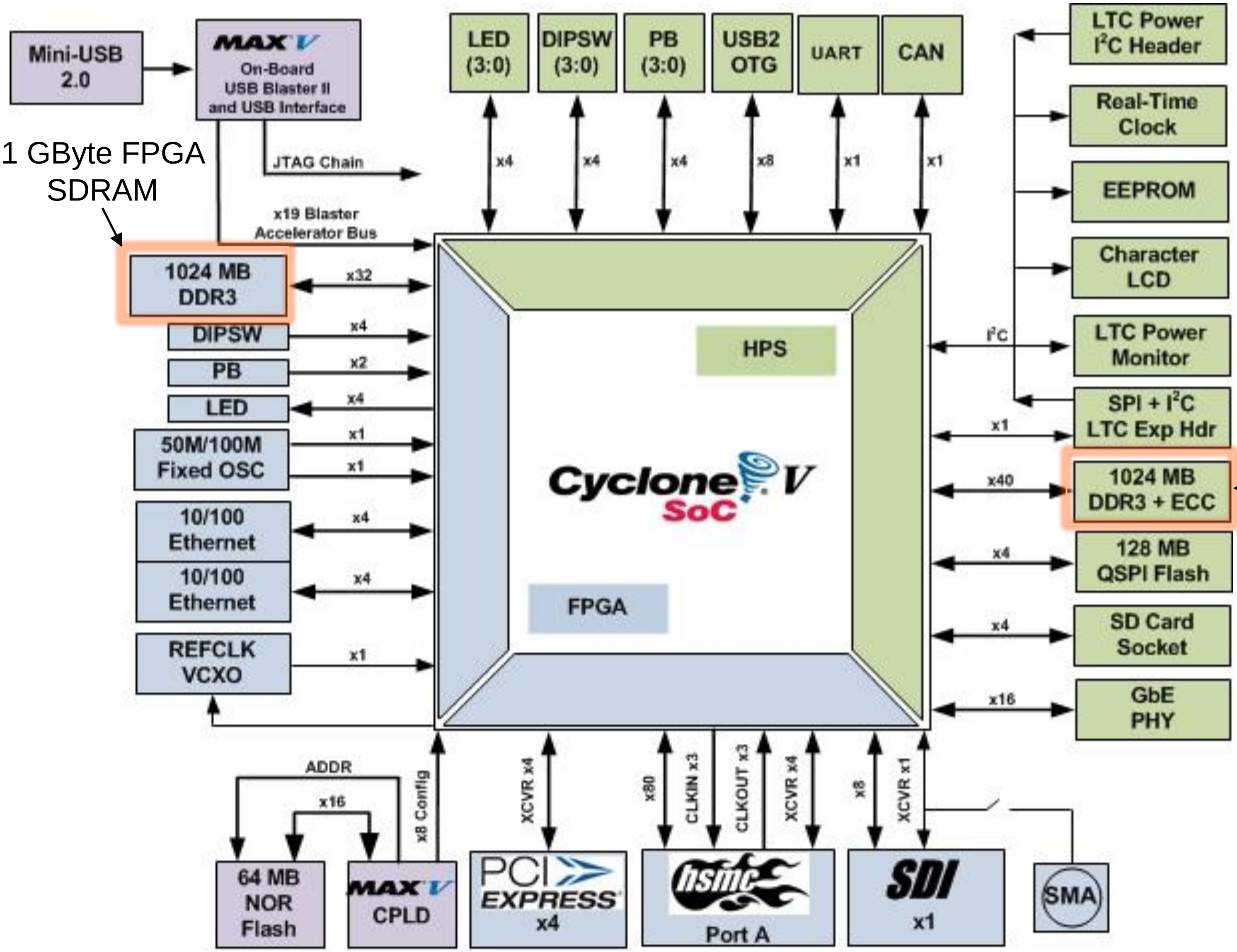


Physical Address Mapping – FPGA to HPS

- FPGA to HPS (F2H) masters see 4 GByte address space
- F2H bandwidth to SDRAM limited vs. FPGA to SDRAM bridge



Cyclone V SoC Memory Map Example - SDRAM



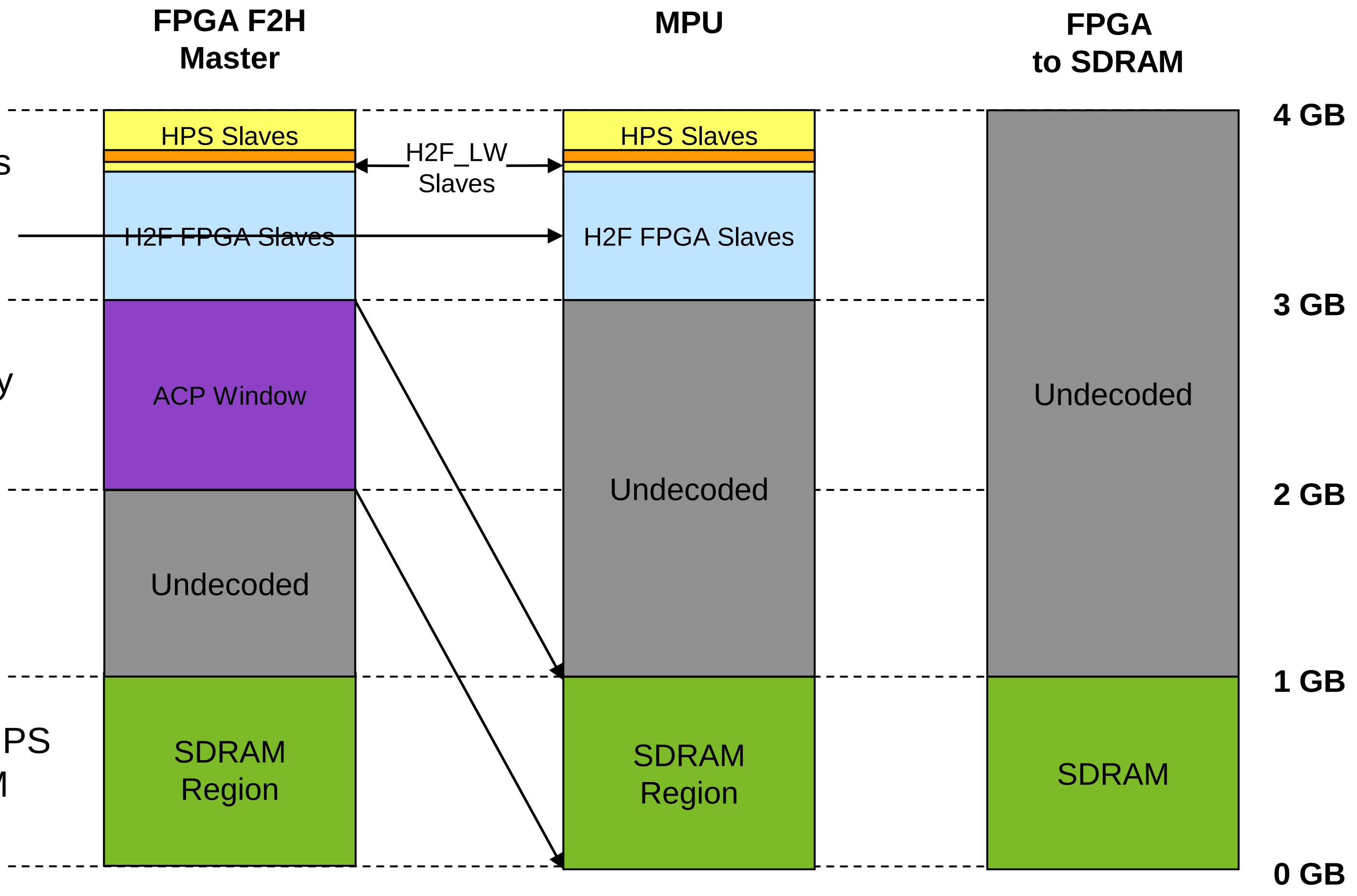
1 GByte FPGA SDRAM

1 GByte HPS SDRAM

Cyclone V SoC GSRD Memory Map Example - SDRAM

- 960MB available for all FPGA slaves
- MPU cannot directly access full 1GByte of FPGA SDRAM
- Not allocatable by kernel

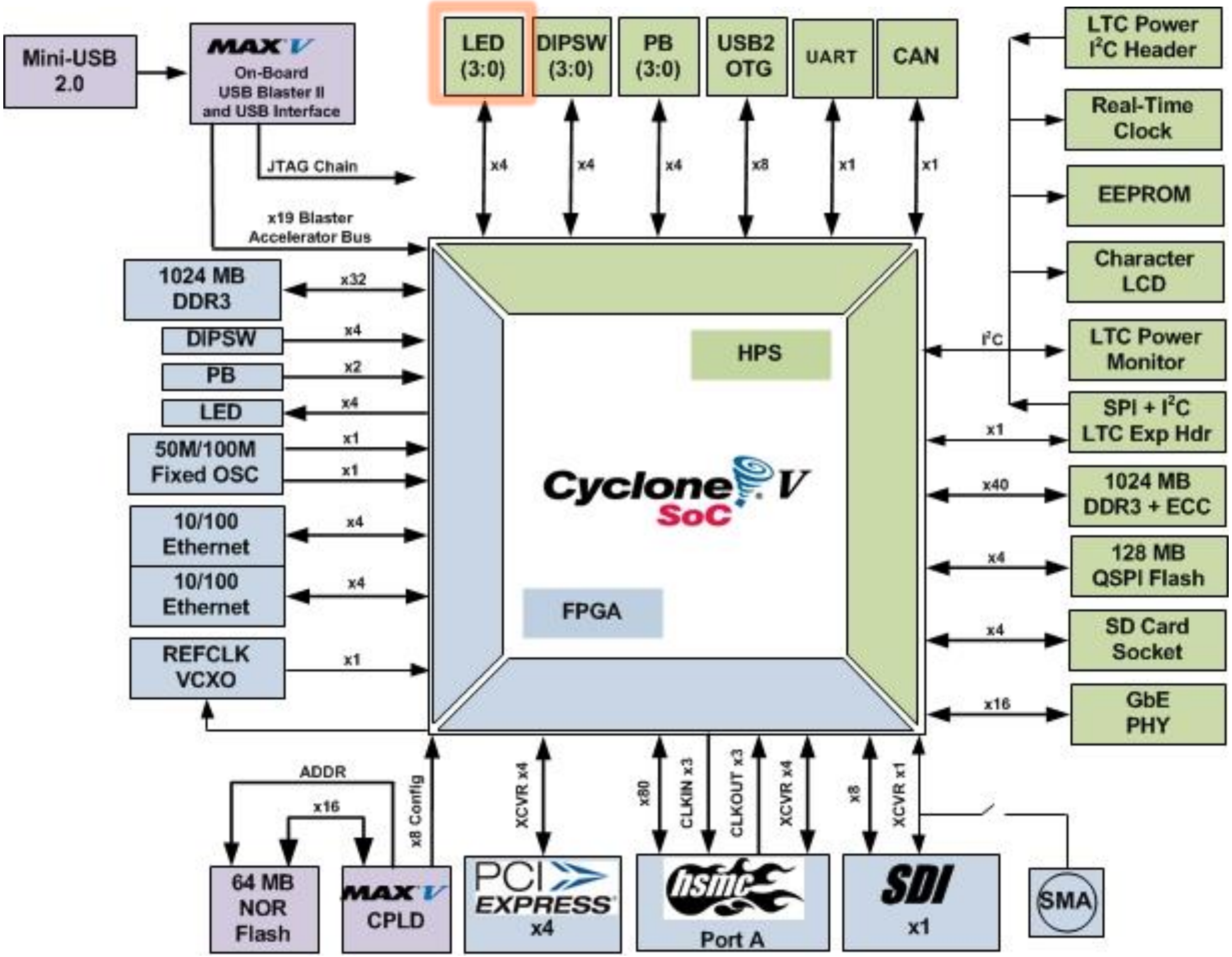
1 GByte HPS SDRAM



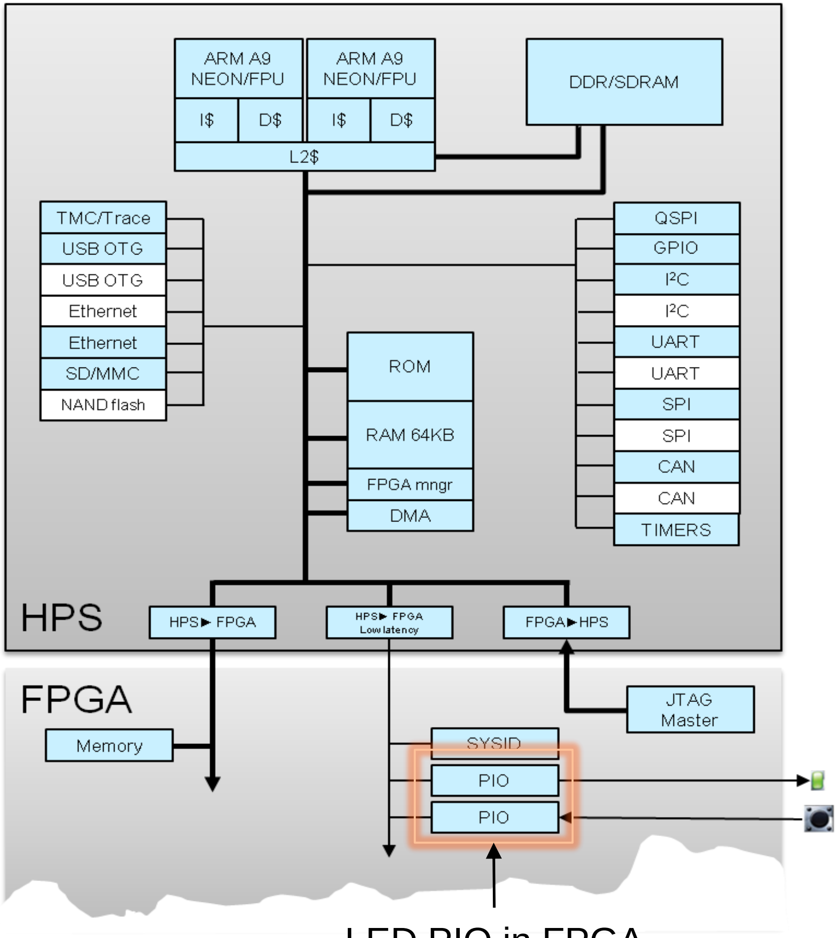
Default remap to 0x0

Cyclone V SoC Memory Map Example - PIO

LED PIO



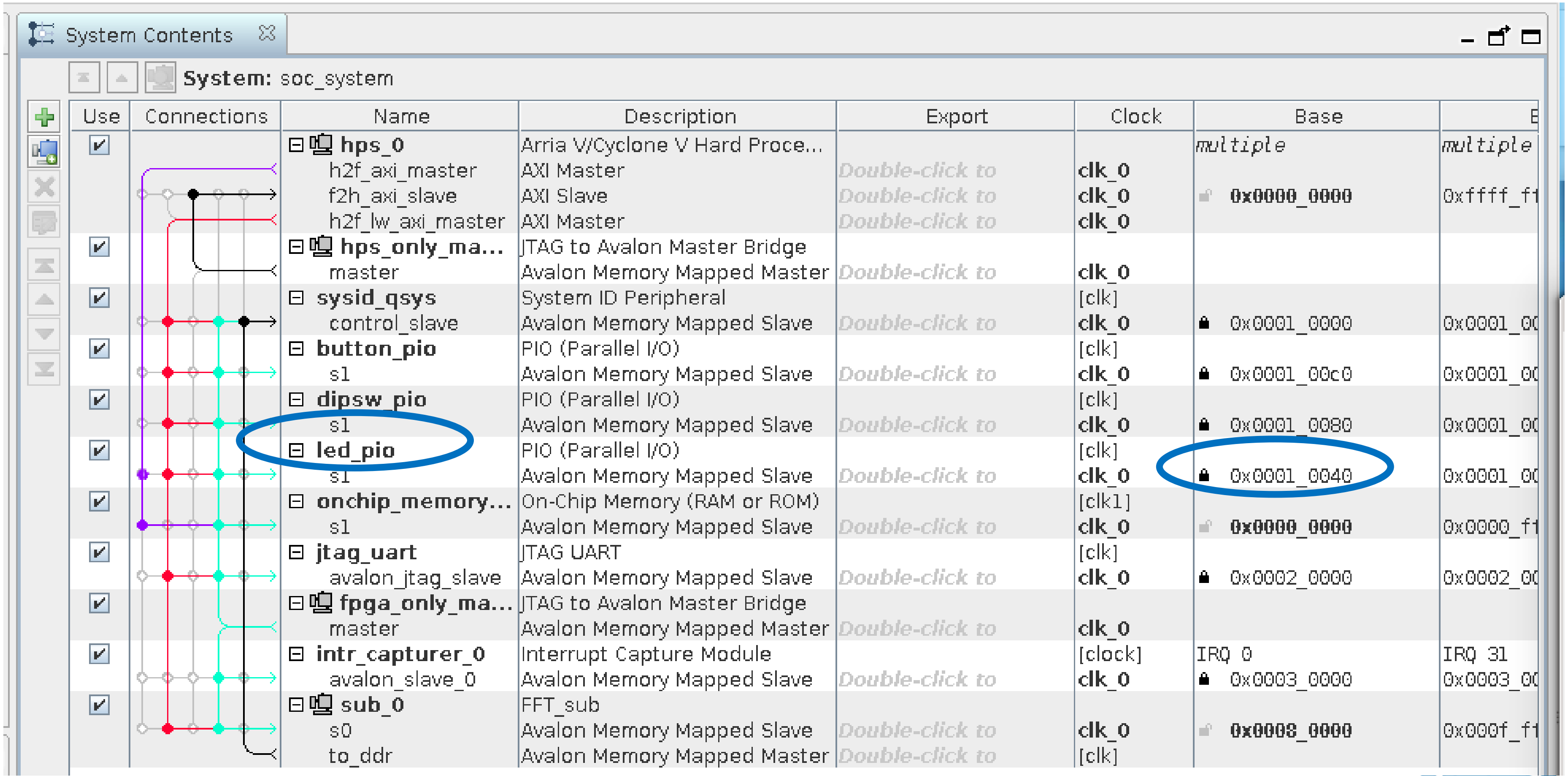
Cyclone V SoC Memory Map Example - PIO



LED PIO in FPGA

Cyclone V SoC GSRD Memory Map Example

- Look at led_pio at Address 0x0001_0040



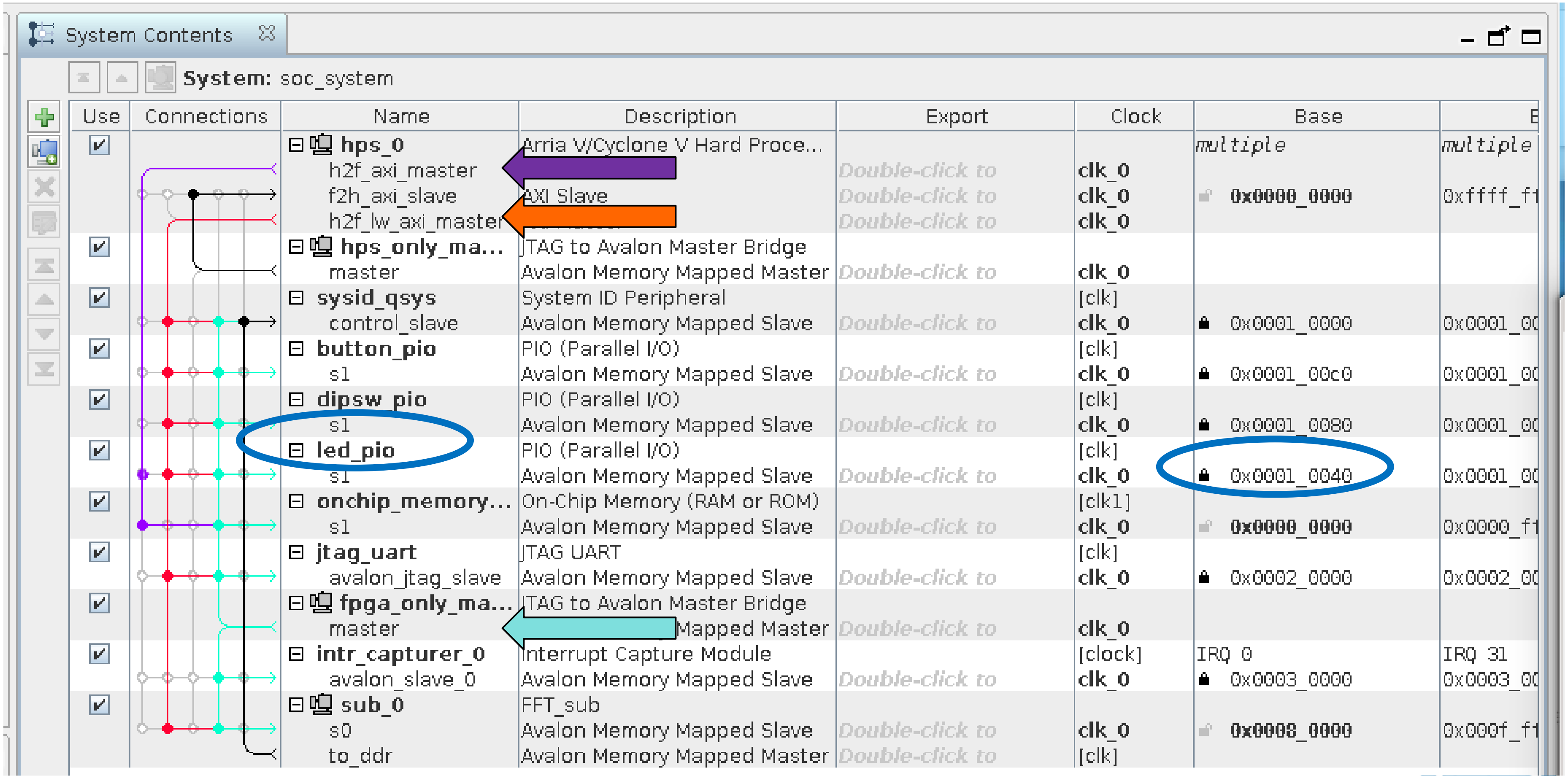
System Contents

System: soc_system

Use	Connections	Name	Description	Export	Clock	Base	E
☑		⊟ hps_0	Arria V/Cyclone V Hard Proce...			multiple	multiple
		h2f_axi_master	AXI Master	Double-click to	clk_0		
		f2h_axi_slave	AXI Slave	Double-click to	clk_0	🔓 0x0000_0000	0xffff_ff
		h2f_lw_axi_master	AXI Master	Double-click to	clk_0		
☑		⊟ hps_only_ma...	JTAG to Avalon Master Bridge				
		master	Avalon Memory Mapped Master	Double-click to	clk_0		
☑		⊟ sysid_qsys	System ID Peripheral		[clk]		
		control_slave	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_0000	0x0001_00
☑		⊟ button_pio	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_00c0	0x0001_00
☑		⊟ dipsw_pio	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_0080	0x0001_00
☑		⊟ led_pio	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_0040	0x0001_00
☑		⊟ onchip_memory...	On-Chip Memory (RAM or ROM)		[clk1]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔓 0x0000_0000	0x0000_ff
☑		⊟ jtag_uart	JTAG UART		[clk]		
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0002_0000	0x0002_00
☑		⊟ fpga_only_ma...	JTAG to Avalon Master Bridge				
		master	Avalon Memory Mapped Master	Double-click to	clk_0		
☑		⊟ intr_capturer_0	Interrupt Capture Module		[clock]	IRQ 0	IRQ 31
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0003_0000	0x0003_00
☑		⊟ sub_0	FFT_sub				
		s0	Avalon Memory Mapped Slave	Double-click to	clk_0	🔓 0x0008_0000	0x000f_ff
		to_ddr	Avalon Memory Mapped Master	Double-click to	[clk]		

Cyclone V SoC GSRD Memory Map Example

- It is connected to 3 Masters.



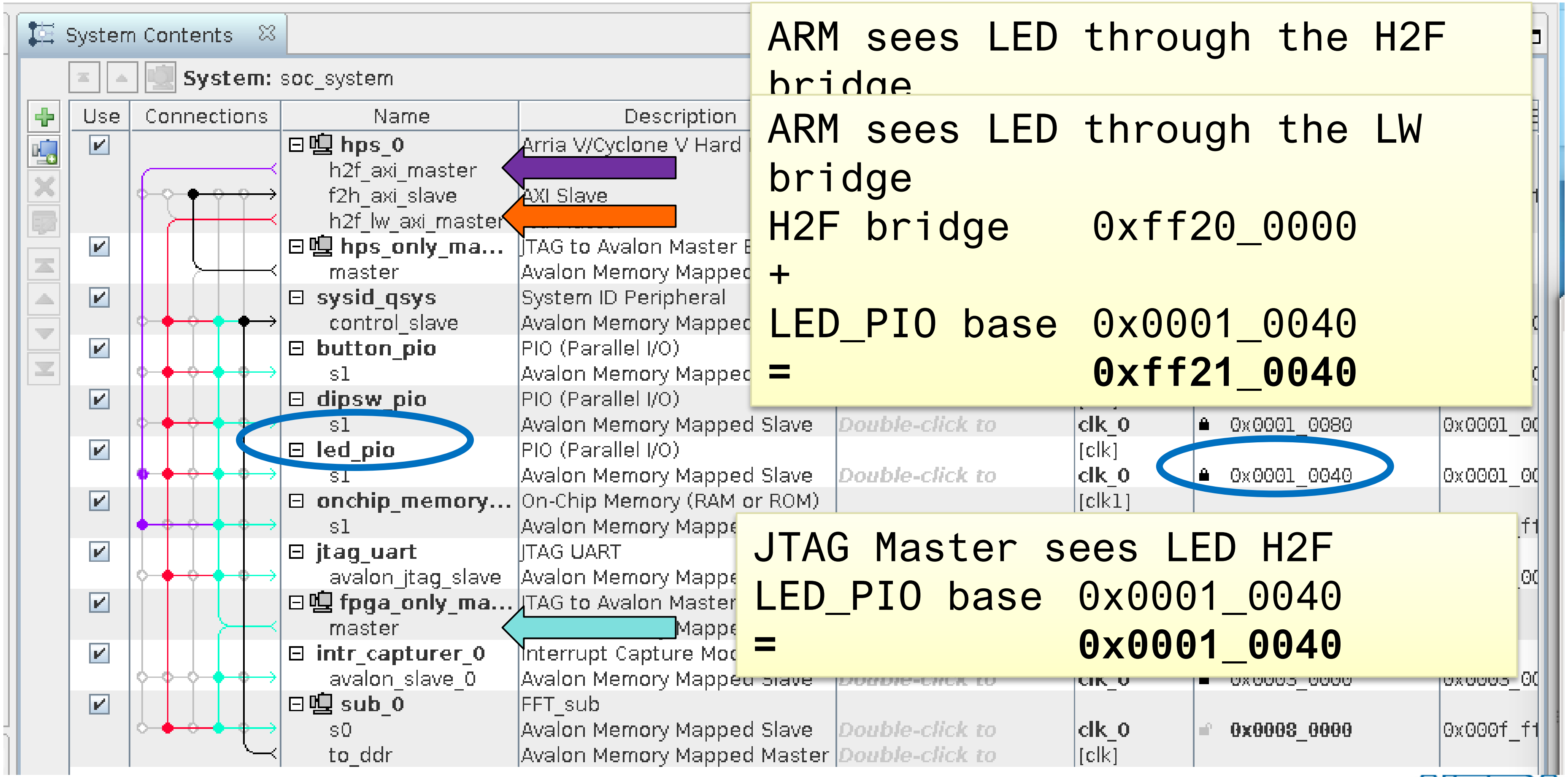
System Contents

System: soc_system

Use	Connections	Name	Description	Export	Clock	Base	E
☑		⊟ **hps_0**	Arria V/Cyclone V Hard Proce...			*multiple*	*multiple*
		h2f_axi_master		Double-click to	clk_0		
		f2h_axi_slave	AXI Slave	Double-click to	clk_0	**0x0000_0000**	0xffff_ff
		h2f_lw_axi_master		Double-click to	clk_0		
☑		⊟ **hps_only_ma...**	JTAG to Avalon Master Bridge				
		master	Avalon Memory Mapped Master	Double-click to	clk_0		
☑		⊟ **sysid_qsys**	System ID Peripheral		[clk]		
		control_slave	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0001_0000	0x0001_00
☑		⊟ **button_pio**	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0001_00c0	0x0001_00
☑		⊟ **dipsw_pio**	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0001_0080	0x0001_00
☑		⊟ **led_pio**	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0001_0040	0x0001_00
☑		⊟ **onchip_memory...**	On-Chip Memory (RAM or ROM)		[clk1]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	**0x0000_0000**	0x0000_ff
☑		⊟ **jtag_uart**	JTAG UART		[clk]		
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0002_0000	0x0002_00
☑		⊟ **fpga_only_ma...**	JTAG to Avalon Master Bridge				
		master	Mapped Master	Double-click to	clk_0		
☑		⊟ **intr_capturer_0**	Interrupt Capture Module		[clock]	IRQ 0	IRQ 31
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0003_0000	0x0003_00
☑		⊟ **sub_0**	FFT_sub				
		s0	Avalon Memory Mapped Slave	Double-click to	clk_0	**0x0008_0000**	0x000f_ff
		to_ddr	Avalon Memory Mapped Master	Double-click to	[clk]		

Cyclone V SoC GSRD Memory Map Example

- Each master sees the slave at a different address
- These address are offsets from the HPS bridge address



System Contents

System: soc_system

Use	Connections	Name	Description				
☑		⊟ hps_0	Arria V/Cyclone V Hard				
		h2f_axi_master					
		f2h_axi_slave	AXI Slave				
		h2f_lw_axi_master					
☑		⊟ hps_only_ma...	JTAG to Avalon Master E				
		master	Avalon Memory Mappec				
☑		⊟ sysid_qsys	System ID Peripheral				
		control_slave	Avalon Memory Mappec				
☑		⊟ button_pio	PIO (Parallel I/O)				
		s1	Avalon Memory Mappec				
☑		⊟ dipsw_pio	PIO (Parallel I/O)				
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_0080	0x0001_00
☑		⊟ led_pio	PIO (Parallel I/O)		[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to	clk_0	🔒 0x0001_0040	0x0001_00
☑		⊟ onchip_memory...	On-Chip Memory (RAM or ROM)		[clk1]		
		s1	Avalon Memory Mappe				f1
☑		⊟ jtag_uart	JTAG UART				
		avalon_jtag_slave	Avalon Memory Mappe				00
☑		⊟ fpga_only_ma...	JTAG to Avalon Master				
		master	Mappe				
☑		⊟ intr_capturer_0	Interrupt Capture Moc				
		avalon_slave_0	Avalon Memory Mapped				
☑		⊟ sub_0	FFT_sub				
		s0	Avalon Memory Mapped Slave	Double-click to	clk_0	0x0008_0000	0x000f_ff
		to_ddr	Avalon Memory Mapped Master	Double-click to	[clk]		

ARM sees LED through the H2F bridge

ARM sees LED through the LW bridge

```
H2F bridge     0xff20_0000
+
LED_PIO base   0x0001_0040
=              0xff21_0040
```

JTAG Master sees LED H2F

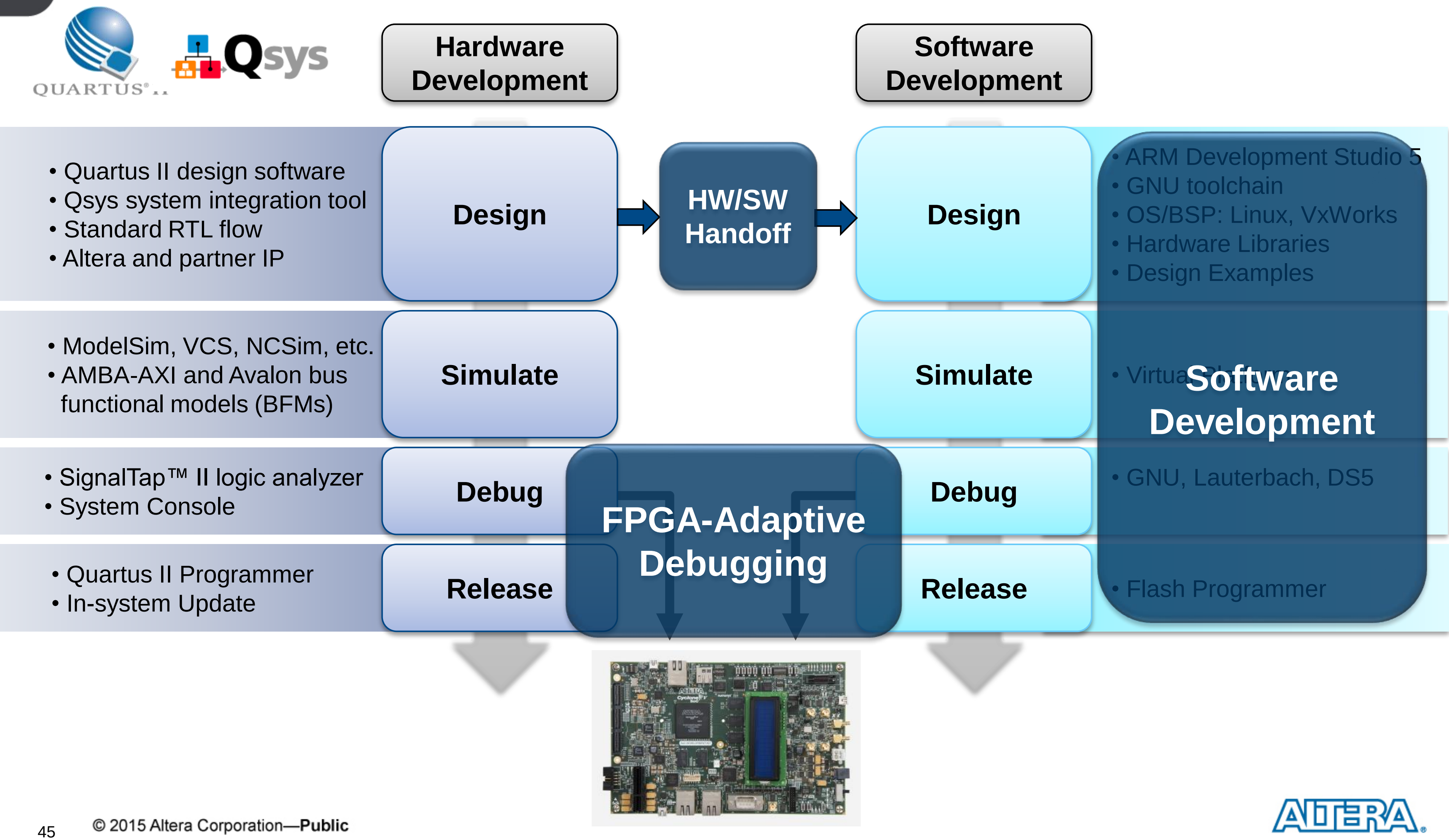
```
LED_PIO base  0x0001_0040
=             0x0001_0040
```

SoCFPGA Development Flow & Tools

Why Does This Matter for Linux Development?

- Altera SoCs offer unique advantages
  - User specified peripheral set
  - Tightly-coupled MPU & FPGA fabric
  - One of a kind HW/SW debug capabilities
- Altera SoCs have unique requirements
  - Understanding of paths to exchange data between MPU and FPGA
  - Building custom BSP for user specified peripheral set
  - Correct handling and configuration of HPS/FPGA bridges
- Altera SoCs have tools to enable the power of a integrated MPU and FPGA

Altera SoC Embedded Design Suite



FPGA Design Flow

Software Design Flow

Hardware Development

- Quartus II design software
- Qsys system integration tool
- Standard RTL flow
- Altera and partner IP

Design → **HW/SW Handoff** → **Design**

- ModelSim, VCS, NCSim, etc.
- AMBA-AXI and Avalon bus functional models (BFMs)

Simulate

- SignalTap™ II logic analyzer
- System Console

Debug

- Quartus II Programmer
- In-system Update

Release

FPGA-Adaptive Debugging

Software Development

Design

- ARM Development Studio 5
- GNU toolchain
- OS/BSP: Linux, VxWorks
- Hardware Libraries
- Design Examples

Simulate

- Virtua

Debug

- GNU, Lauterbach, DS5

Release

- Flash Programmer

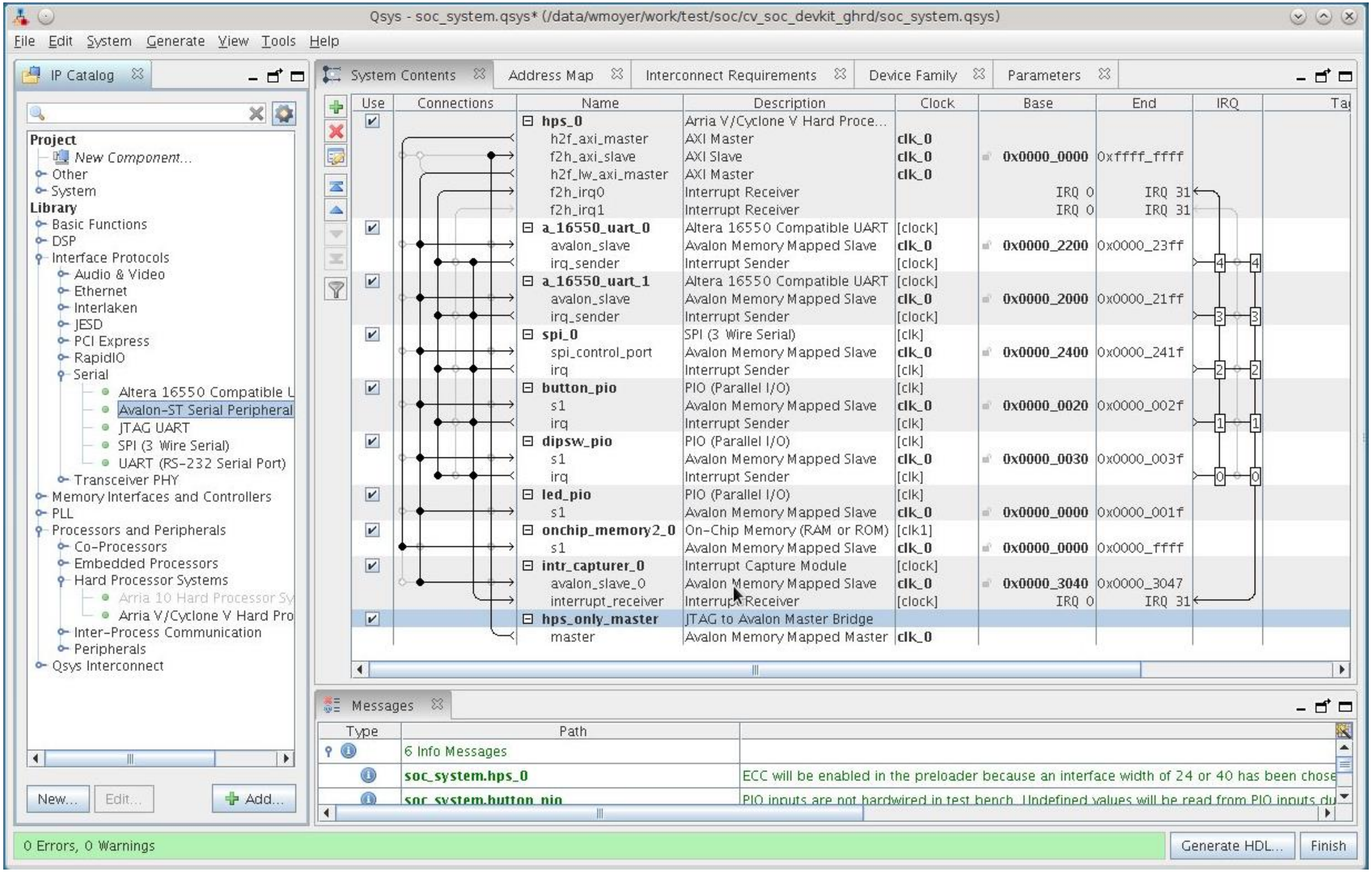
Software Development

So… what exactly is Qsys?

GUI based system integration tool for HW system design using IP blocks.

- Simplifies complex system development
- Raises the level of design abstraction
- Provides a standard platform:
  - IP integration
  - Custom IP authoring
  - IP verification
- Enables design re-use
- Scales easily to meet the needs of end product
- Reduces time to market

Qsys System Integration Platform



Qsys - soc_system.qsys* (/data/wmoyer/work/test/soc/cv_soc_devkit_ghrd/soc_system.qsys)

File Edit System Generate View Tools Help
IP Catalog
System Contents
Address Map
Interconnect Requirements
Device Family
Parameters

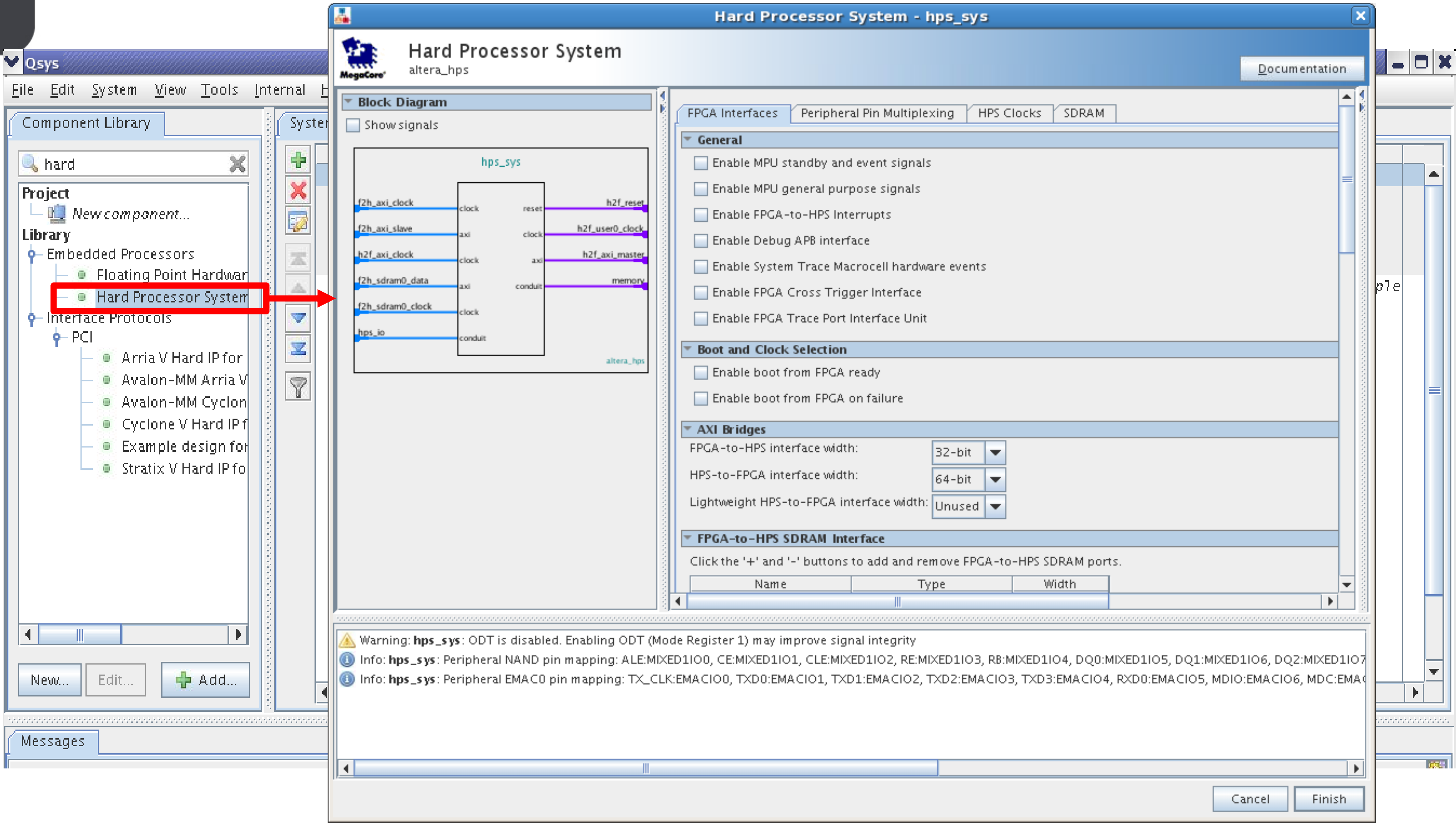
- Project
  - New Component...
  - Other
  - System
- Library
  - Basic Functions
  - DSP
  - Interface Protocols
    - Audio & Video
    - Ethernet
    - Interlaken
    - JESD
    - PCI Express
    - RapidIO
    - Serial
      - Altera 16550 Compatible U
      - Avalon-ST Serial Peripheral
      - JTAG UART
      - SPI (3 Wire Serial)
      - UART (RS-232 Serial Port)
    - Transceiver PHY
  - Memory Interfaces and Controllers
  - PLL
  - Processors and Peripherals
    - Co-Processors
    - Embedded Processors
    - Hard Processor Systems
      - Arria 10 Hard Processor Sy
      - Arria V/Cyclone V Hard Pro
    - Inter-Process Communication
    - Peripherals
  - Qsys Interconnect

Use	Connections	Name	Description	Clock	Base	End	IRQ	Ta
☑		⊟ **hps_0**	Arria V/Cyclone V Hard Proce...					
		h2f_axi_master	AXI Master	**clk_0**				
		f2h_axi_slave	AXI Slave	**clk_0**	**0x0000_0000**	0xffff_ffff		
		h2f_lw_axi_master	AXI Master	**clk_0**				
		f2h_irq0	Interrupt Receiver		IRQ 0	IRQ 31		
		f2h_irq1	Interrupt Receiver		IRQ 0	IRQ 31		
☑		⊟ **a_16550_uart_0**	Altera 16550 Compatible UART	[clock]				
		avalon_slave	Avalon Memory Mapped Slave	**clk_0**	**0x0000_2200**	0x0000_23ff		
		irq_sender	Interrupt Sender	[clock]			4	
☑		⊟ **a_16550_uart_1**	Altera 16550 Compatible UART	[clock]				
		avalon_slave	Avalon Memory Mapped Slave	**clk_0**	**0x0000_2000**	0x0000_21ff		
		irq_sender	Interrupt Sender	[clock]			3	
☑		⊟ **spi_0**	SPI (3 Wire Serial)	[clk]				
		spi_control_port	Avalon Memory Mapped Slave	**clk_0**	**0x0000_2400**	0x0000_241f		
		irq	Interrupt Sender	[clk]			2	
☑		⊟ **button_pio**	PIO (Parallel I/O)	[clk]				
		s1	Avalon Memory Mapped Slave	**clk_0**	**0x0000_0020**	0x0000_002f		
		irq	Interrupt Sender	[clk]			1	
☑		⊟ **dipsw_pio**	PIO (Parallel I/O)	[clk]				
		s1	Avalon Memory Mapped Slave	**clk_0**	**0x0000_0030**	0x0000_003f		
		irq	Interrupt Sender	[clk]			0	
☑		⊟ **led_pio**	PIO (Parallel I/O)	[clk]				
		s1	Avalon Memory Mapped Slave	**clk_0**	**0x0000_0000**	0x0000_001f		
☑		⊟ **onchip_memory2_0**	On-Chip Memory (RAM or ROM)	[clk1]				
		s1	Avalon Memory Mapped Slave	**clk_0**	**0x0000_0000**	0x0000_ffff		
☑		⊟ **intr_capturer_0**	Interrupt Capture Module	[clock]				
		avalon_slave_0	Avalon Memory Mapped Slave	**clk_0**	**0x0000_3040**	0x0000_3047		
		interrupt_receiver	Interrupt Receiver	[clock]	IRQ 0	IRQ 31		
☑		⊟ **hps_only_master**	JTAG to Avalon Master Bridge					
		master	Avalon Memory Mapped Master	**clk_0**				

Type	Path	
ⓘ	6 Info Messages	
ⓘ	**soc_system.hps_0**	ECC will be enabled in the preloader because an interface width of 24 or 40 has been chose
ⓘ	**soc_system.button_pio**	PIO inputs are not hardwired in test bench. Undefined values will be read from PIO inputs du

0 Errors, 0 Warnings

Hard Processor System Configuration



Hard Processor System

altera_hps

FPGA Interfaces | Peripheral Pin Multiplexing | HPS Clocks | SDRAM

General

- □ Enable MPU standby and event signals
- □ Enable MPU general purpose signals
- □ Enable FPGA-to-HPS Interrupts
- □ Enable Debug APB interface
- □ Enable System Trace Macrocell hardware events
- □ Enable FPGA Cross Trigger Interface
- □ Enable FPGA Trace Port Interface Unit

Boot and Clock Selection

- □ Enable boot from FPGA ready
- □ Enable boot from FPGA on failure

AXI Bridges

FPGA-to-HPS interface width: 32-bit

HPS-to-FPGA interface width: 64-bit

Lightweight HPS-to-FPGA interface width: Unused

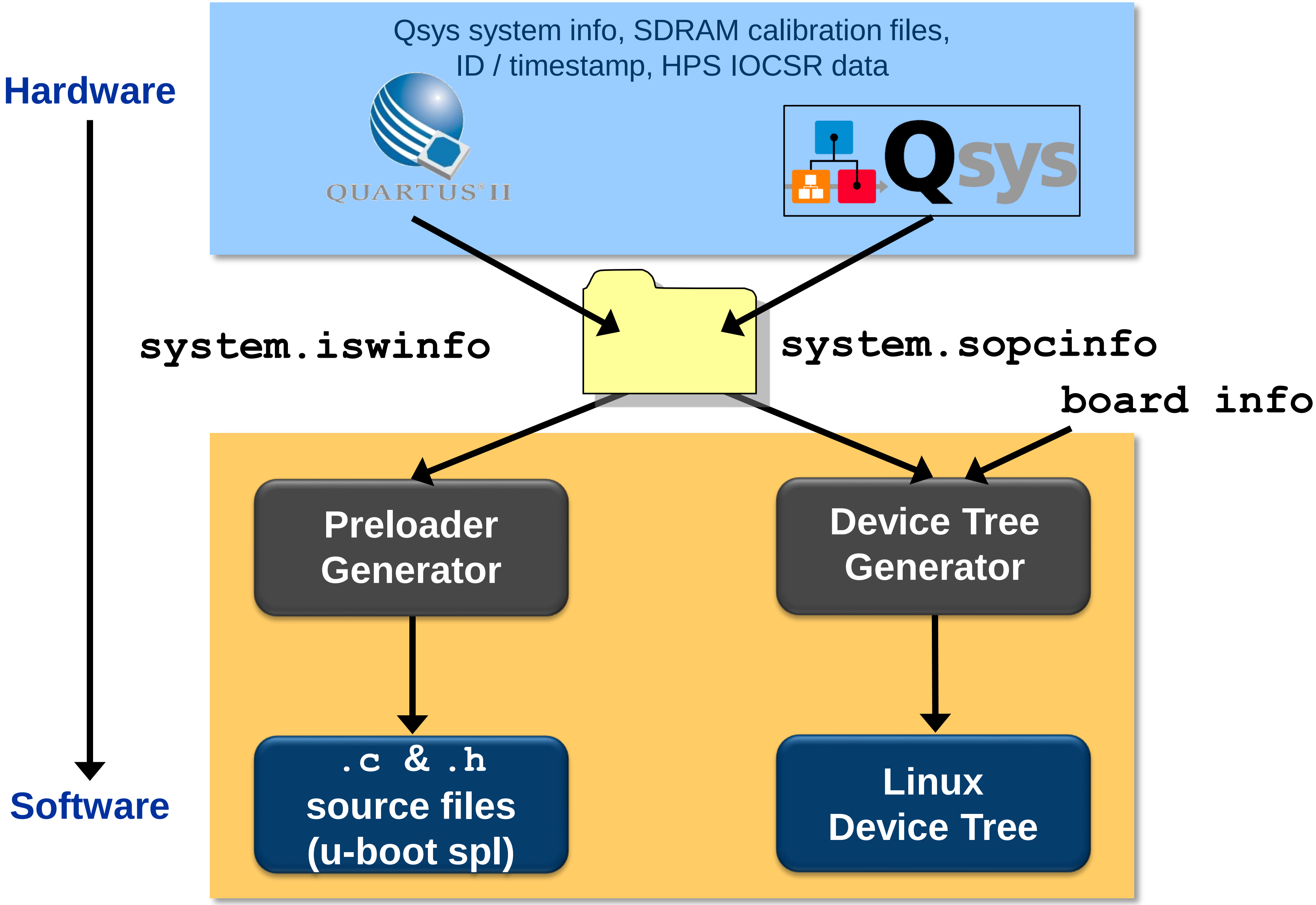
FPGA-to-HPS SDRAM Interface

Click the '+' and '-' buttons to add and remove FPGA-to-HPS SDRAM ports.

Name	Type	Width

```
Warning: hps_sys: ODT is disabled. Enabling ODT (Mode Register 1) may improve signal integrity
Info: hps_sys: Peripheral NAND pin mapping: ALE:MIXED1IO0, CE:MIXED1IO1, CLE:MIXED1IO2, RE:MIXED1IO3, RB:MIXED1IO4, DQ0:MIXED1IO5, DQ1:MIXED1IO6, DQ2:MIXED1IO7
Info: hps_sys: Peripheral EMAC0 pin mapping: TX_CLK:EMACIO0, TXD0:EMACIO1, TXD1:EMACIO2, TXD3:EMACIO3, TXD4:EMACIO4, RXD0:EMACIO5, MDIO:EMACIO6, MDC:EMA
```

Linux HW/SW Handoff – Cyclone V SoC and Arria V SoC

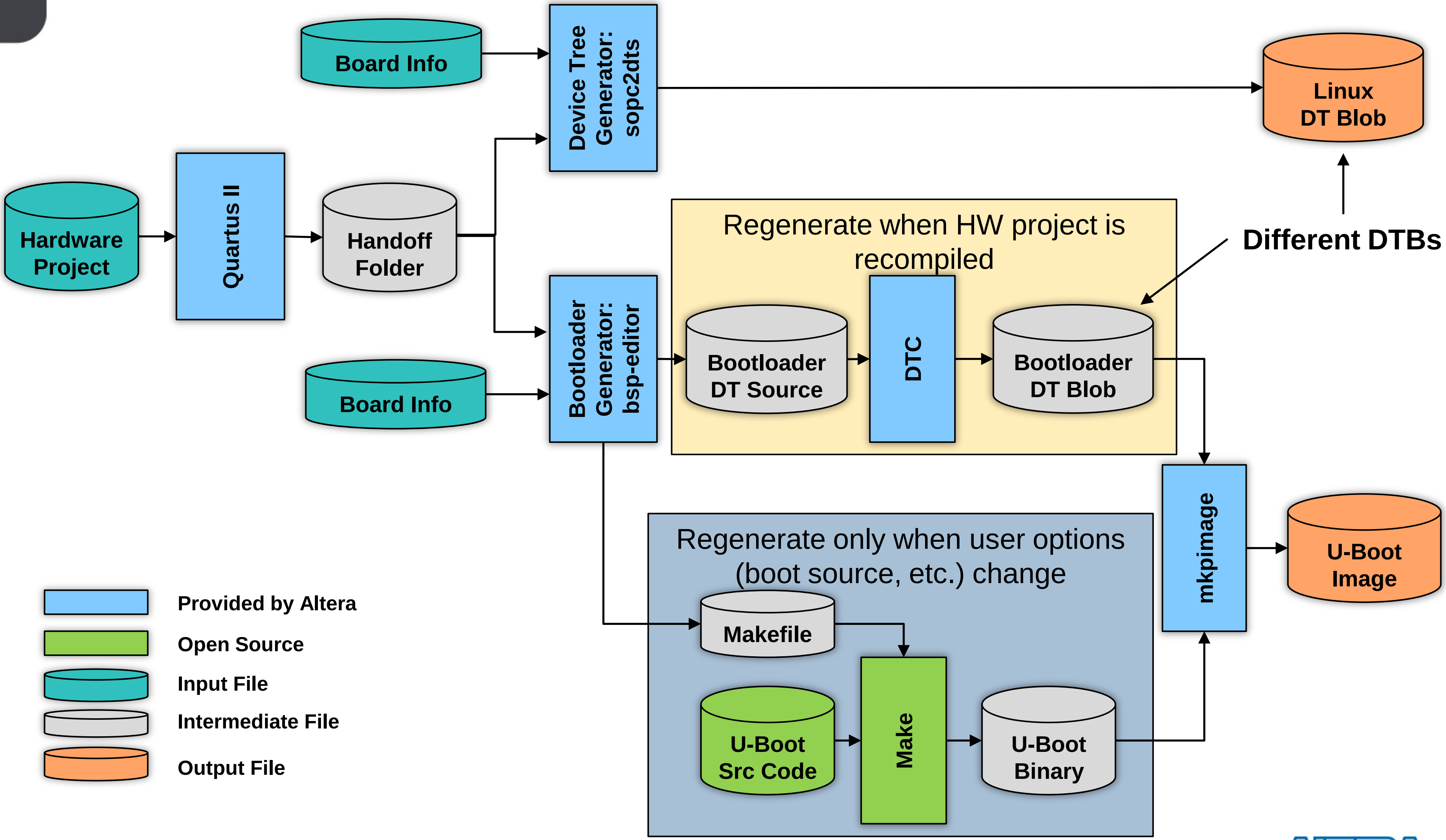


Hardware

Software

Qsys system info, SDRAM calibration files,
ID / timestamp, HPS IOCSR data

Linux HW/SW Handoff – Arria 10 SoC

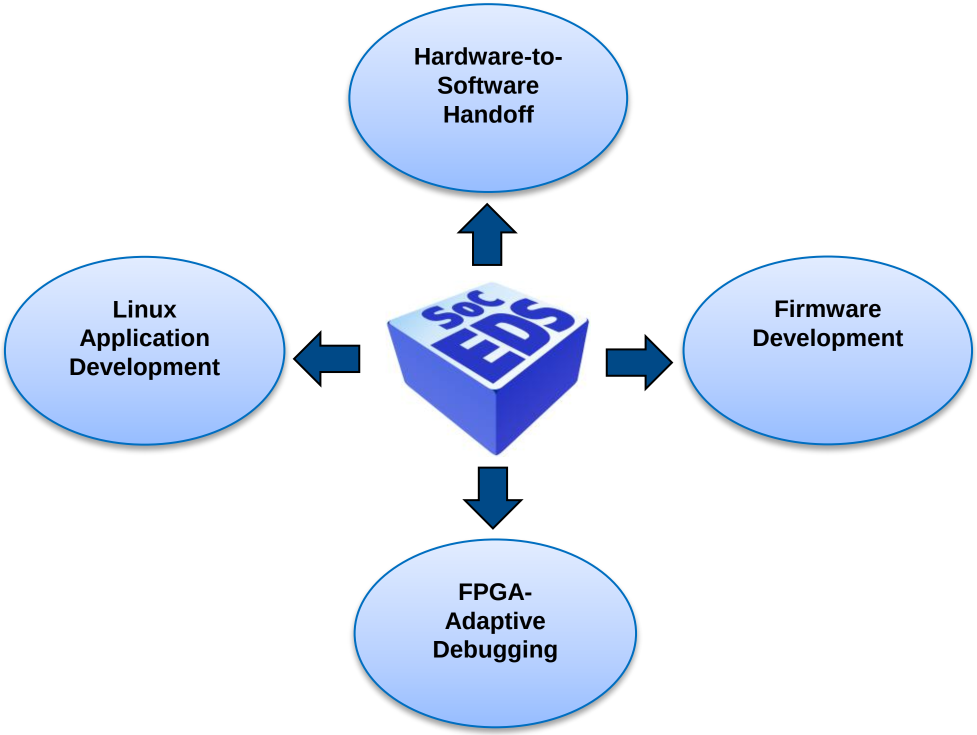


- 🟦 Provided by Altera
- 🟩 Open Source
- Input File
- Intermediate File
- 🟧 Output File

Altera SoC Embedded Design Suite

Comprehensive Suite SW Dev Tools

- Hardware / software handoff tools
  - Preloader & Device Tree Generators
- Bare-metal application development
  - SoC Hardware Libraries
  - Bare-metal compiler tools
- FPGA-adaptive debugging
  - ARM DS-5 Altera Edition Toolkit
- Linux application development
  - Yocto Linux build environment
  - Pre-built binaries for Linux / U-Boot
  - Work in conjunction with the Community Portal
- Design examples

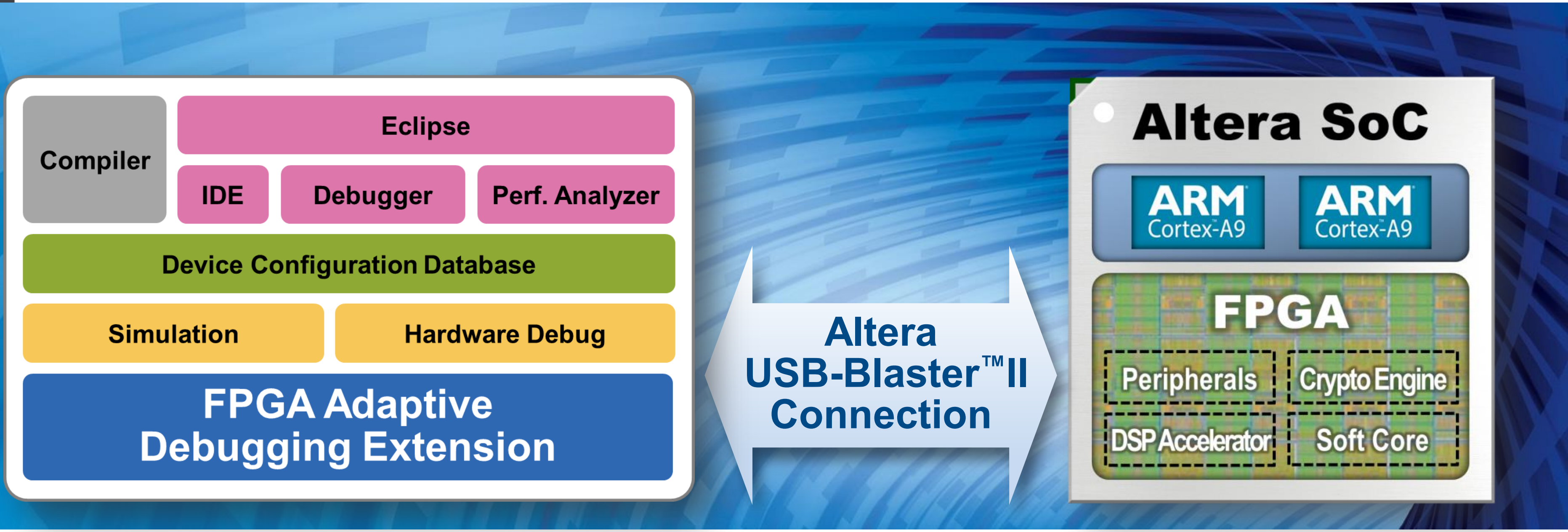


✓ ***Free Web Edition***

✓ ***Subscription Edition***

✓ ***Free 30-day Eval***

Industry First: FPGA-Adaptive Debugging

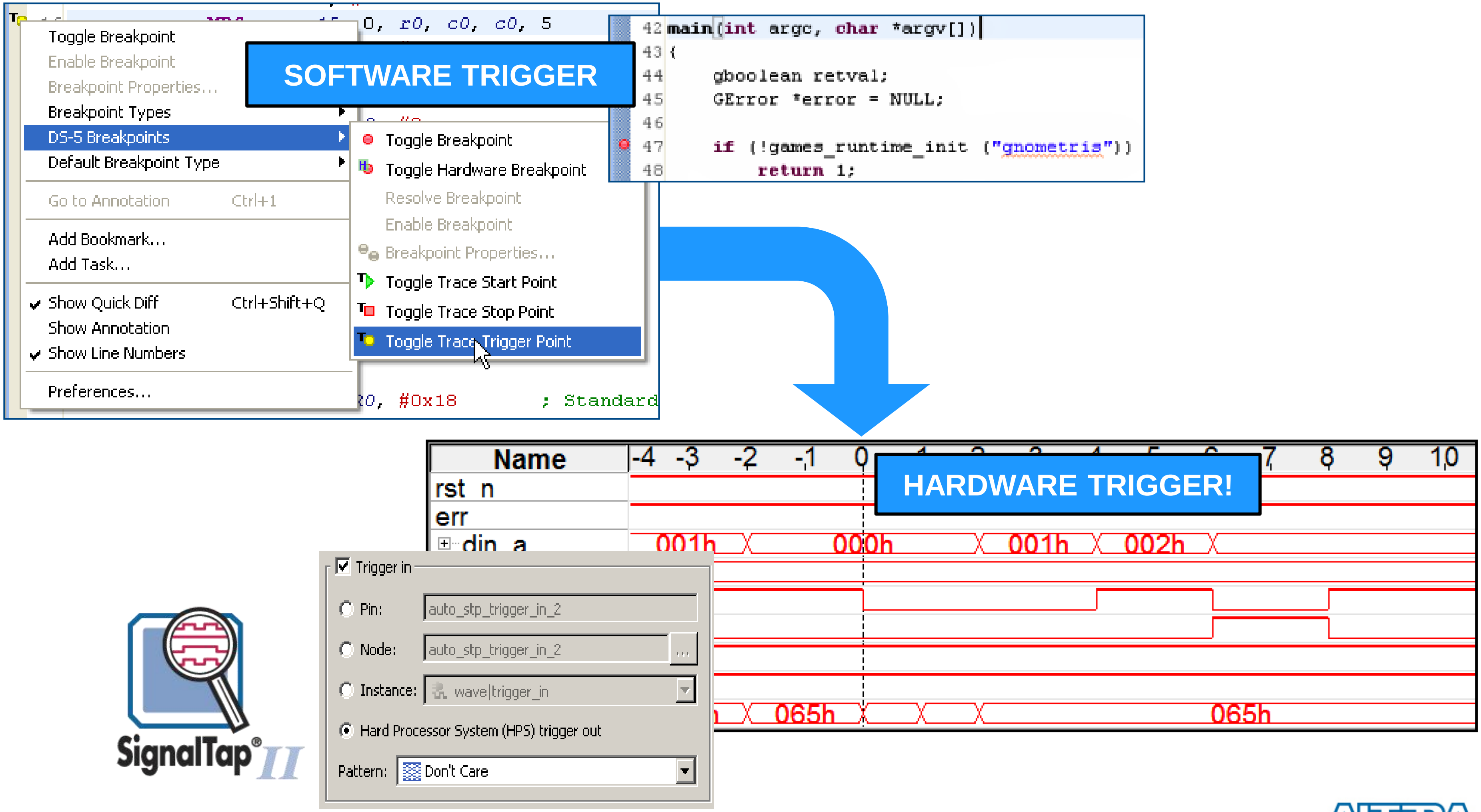


ARM® Development Studio 5 (DS-5™) Altera® Edition Toolkit

- Removes debugging barrier between CPUs and FPGA
- Exclusive OEM agreement between Altera and ARM
- Result of innovation in silicon, software, and business model
- Supports FPGA-Adaptive Linux kernel, driver & application debug

Cross-Domain Debug 1

- Trigger from software world to FPGA world



SOFTWARE TRIGGER

```
main(int argc, char *argv[])
{
    gboolean retval;
    GError *error = NULL;

    if (!games_runtime_init ("gnometris"))
        return 1;
```

HARDWARE TRIGGER!

☑ Trigger in

○ Pin: auto_stp_trigger_in_2

○ Node: auto_stp_trigger_in_2

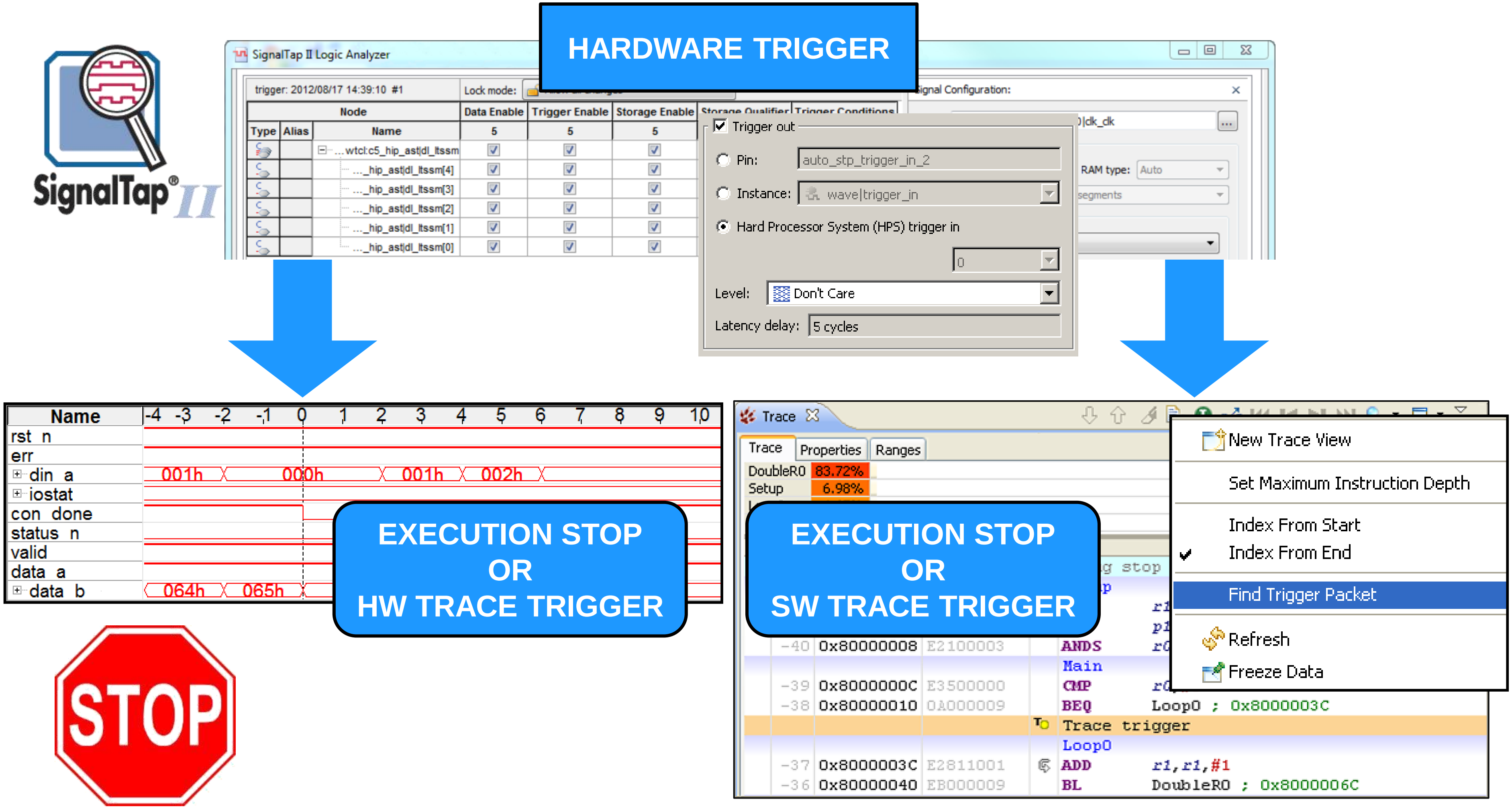
○ Instance: wave|trigger_in

● Hard Processor System (HPS) trigger out

Pattern: Don't Care

Cross-Domain Debug 2

- Trigger from FPGA world to software world



SignalTap II Logic Analyzer

trigger: 2012/08/17 14:39:10 #1 — Lock mode:

Type	Alias	Name	Data Enable 5	Trigger Enable 5	Storage Enable 5
		...wtcl:c5_hip_ast|dl_ltssm	☑	☑	☑
		..._hip_ast|dl_ltssm[4]	☑	☑	☑
		..._hip_ast|dl_ltssm[3]	☑	☑	☑
		..._hip_ast|dl_ltssm[2]	☑	☑	☑
		..._hip_ast|dl_ltssm[1]	☑	☑	☑
		..._hip_ast|dl_ltssm[0]	☑	☑	☑

Signal Configuration:

☑ Trigger out

○ Pin: auto_stp_trigger_in_2

○ Instance: wave|trigger_in

● Hard Processor System (HPS) trigger in

0

Level: Don't Care

Latency delay: 5 cycles

RAM type: Auto

Trace

Trace | Properties | Ranges

DoubleR0 83.72%

Setup 6.98%

```
-40  0x80000008  E2100003   ANDS
                            Main
-39  0x8000000C  E3500000   CMP
-38  0x80000010  0A000009   BEQ        Loop0 ; 0x8000003C
                            Trace trigger
                            Loop0
-37  0x8000003C  E2811001   ADD        r1,r1,#1
-36  0x80000040  EB000009   BL         DoubleR0 ; 0x8000006C
```

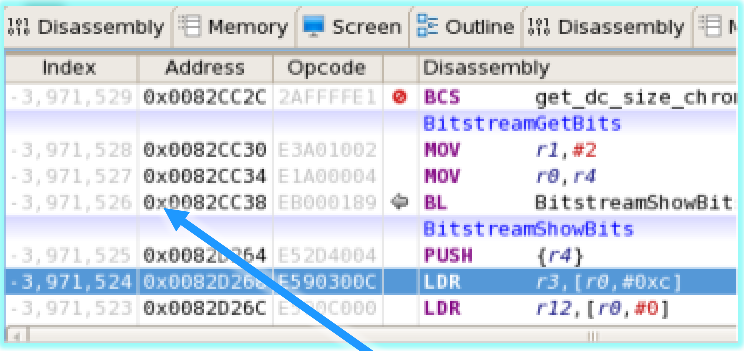
Correlate HW and SW Events

- Debug event trigger point set from either:

  SignalTap™ II Logic Analyzer
  or
  DS-5 debugger

- Captured trace can then be analyzed using timestamp-correlated events

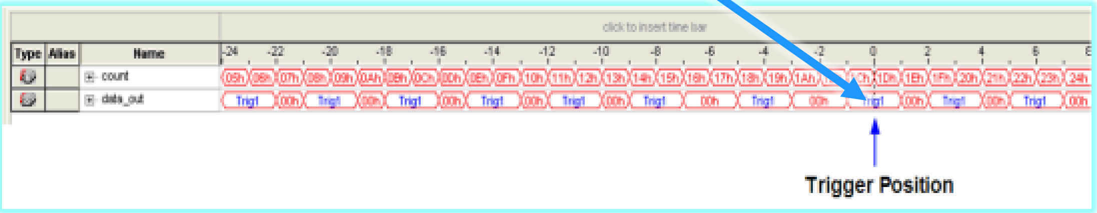
ARM® DS-5™ Toolkit



Index	Address	Opcode		Disassembly
-3,971,529	0x0082CC2C	2AFFFFE1	●	BCS get_dc_size_chro
				BitstreamGetBits
-3,971,528	0x0082CC30	E3A01002		MOV r1,#2
-3,971,527	0x0082CC34	E1A00004		MOV r0,r4
-3,971,526	0x0082CC38	EB000189	⇦	BL BitstreamShowBit
				BitstreamShowBits
-3,971,525	0x0082D264	E52D4004		PUSH {r4}
-3,971,524	0x0082D268	E590300C		LDR r3,[r0,#0xc]
-3,971,523	0x0082D26C	E59C000		LDR r12,[r0,#0]

Timestamp Correlated

SignalTap II Logic Analyzer



click to insert time bar
Type
Alias
Name
count
data_out
-24
-22
-20
-18
-16
-14
-12
-10
-8
-6
-4
-2
0
2
4
6
05h 06h 07h 08h 09h 0Ah 0Bh 0Ch 0Dh 0Eh 0Fh 10h 11h 12h 13h 14h 15h 16h 17h 18h 19h 1Ah 1Ch 1Dh 1Eh 1Fh 20h 21h 22h 23h 24h
Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h Trig1 00h
Trigger Position

SoC EDS Editions Summary

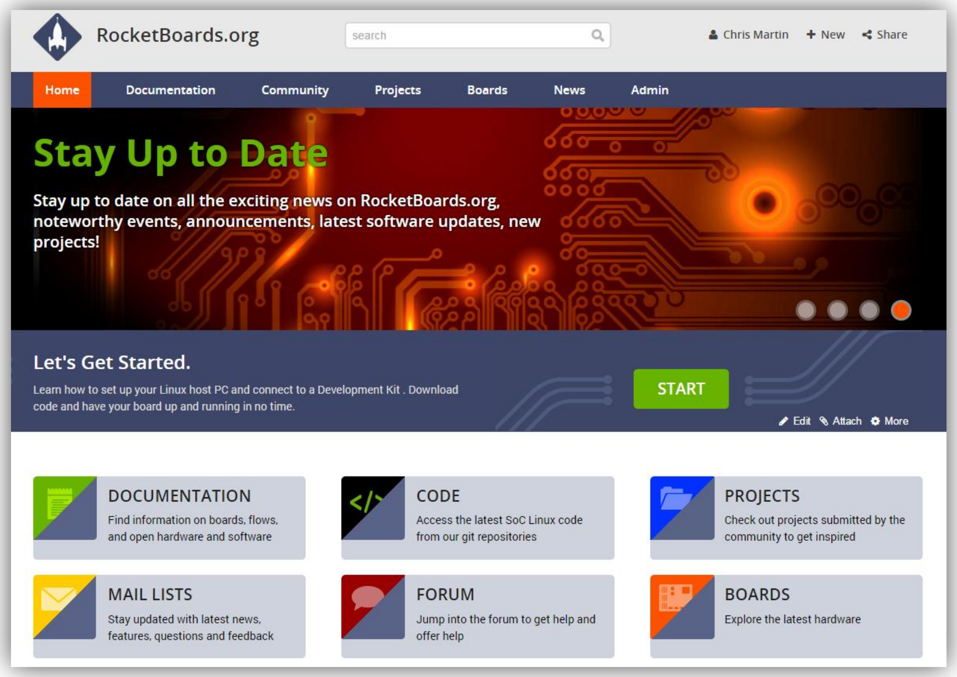
Component	Key Feature	Web Edition	Subscription Edition	30-Day Evaluation
Hardware/Software Handoff Tools	Preloader Image Generator	x	x	x
	Flash Image Creator	x	x	x
	Device Tree Generator (Linux)	x	x	x
ARM DS-5 Altera Edition	Eclipse IDE	x	x	x
	Debugging over Ethernet (Linux)	x	x	x
	Debugging over USB-Blaster II JTAG		x	x
	Automatic FPGA Register Views		x	x
	Hardware Cross-triggering		x	x
	CPU/FPGA Event Correlation		x	x
Compiler Tool Chains	Linaro Tool Chain (Linux)	x	x	x
	CodeBench Lite EABI (Bare-metal)	X	x	x
Hardware Libraries	Bare-metal programming Support	x	x	x
SoC Programming Examples	Golden System Reference Design	x	x	x

Everything needed for Linux development is free & open source

Altera SoC Linux Overview

Linux for Altera SoCs

- High Quality Linux Support
- Modern release strategy
- Multiple Kernel Versions
- Community Enablement



RocketBoards.org

search

Chris Martin   + New   Share

Stay Up to Date

Stay up to date on all the exciting news on RocketBoards.org, noteworthy events, announcements, latest software updates, new projects!

Let's Get Started.

Learn how to set up your Linux host PC and connect to a Development Kit . Download code and have your board up and running in no time.

DOCUMENTATION
Find information on boards, flows, and open hardware and software

CODE
Access the latest SoC Linux code from our git repositories

PROJECTS
Check out projects submitted by the community to get inspired

MAIL LISTS
Stay updated with latest news, features, questions and feedback

FORUM
Jump into the forum to get help and offer help

BOARDS
Explore the latest hardware

LONG TERM SUPPORT INITIATIVE

Ångström

git

Linaro

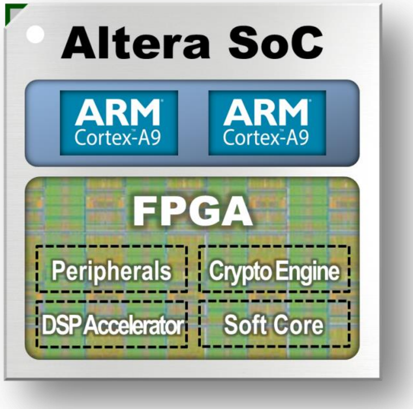
yocto PROJECT

Linux Strategy

- Kernel
  - Same kernel source tree for all SoC’s and NIOS II
    - Same kernel binary for all 32bit SoCFPGA
    - Same kernel binary for all 64bit SoCFPGA
  - Device tree support (SoCs and NIOS II)
  - Upstream and maintain to kernel.org
- U-Boot
  - 2013.01.01 supported for Cyclone V & Arria V SoC
  - 2014.10 supported for Arria 10 SoC
  - 2015.xx support in progress
  - Same U-Boot source tree for all SoC’s and NIOS II
- Toolchain
  - Standard, un-patched Linaro Toolchain:

    gcc-linaro-arm-linux-gnueabihf-4.9-2014.09

Nios® II



Linux Strategy - Build Systems

- Offer support for Angstrom for SoC
  - Embedded Linux distribution, Yocto Project configuration, package manager (OPKG)
    - Uses meta-altera Yocto layer
  - Currently 2014.12
    - Linaro GCC 4.9 for 32 bit SoCs

- Yocto Project Support for SoC
  - SoCFPGA layer (meta-altera) upstreamed to Angstrom
  - Yocto Project v1.7 ready

- Buildroot for both SoC and Nios II
  - Roll your own from relevant source

Build System Resources

- Angstrom flow for SoC

  http://rocketboards.org/foswiki/view/Documentation/AngstromOnSoCFPGA_1

- Yocto flow for SoC

  http://www.rocketboards.org/foswiki/Documentation/YoctoDoraBuildWithMetaAltera

  - Source Poky from the Yocto Project website

    git://git.yoctoproject.org/poky.git

- Buildroot flow for Nios II

  http://rocketboards.org/foswiki/view/Documentation/NiosIILinuxUserManual

- Buildroot flow for SoC

  http://www.rocketboards.org/foswiki/Documentation/BuildrootForSoCFPGA

Angstrom/Yocto Information Resources

- Yocto Project

  https://www.yoctoproject.org/documentation

- Angstrom Distribution

  http://www.angstrom-distribution.org/

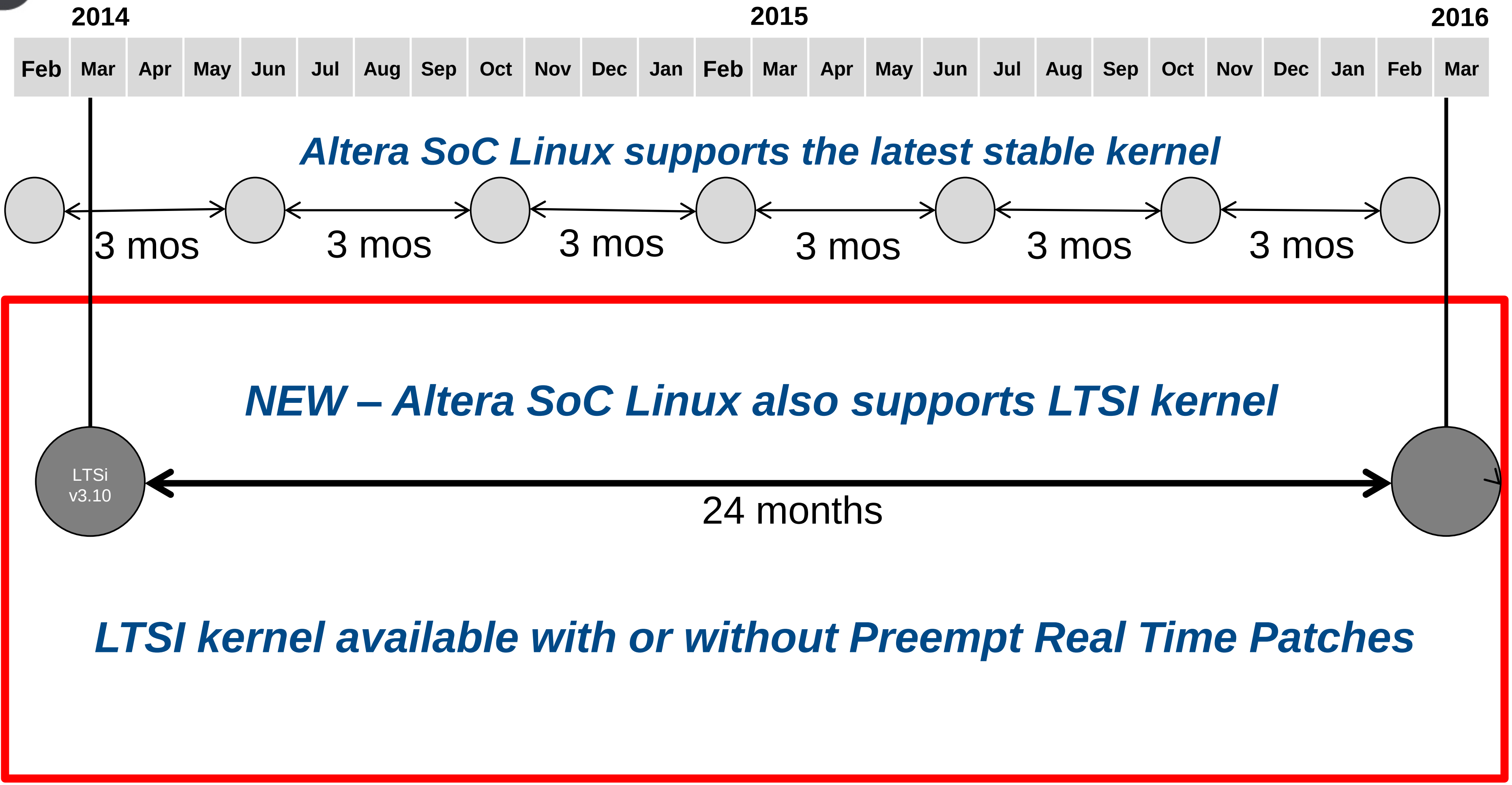
- Open Embedded

  http://www.openembedded.org/wiki/Main_Page

- Building Linux w/ Yocto Linux Foundation class

  LFD405 – Building Embedded Linux with the Yocto Project

Altera SOC Linux Provides Customers Kernel Choices



Altera SoC Linux supports the latest stable kernel

NEW – Altera SoC Linux also supports LTSI kernel

LTSI kernel available with or without Preempt Real Time Patches

Status of the Linux kernel for SoC FPGA

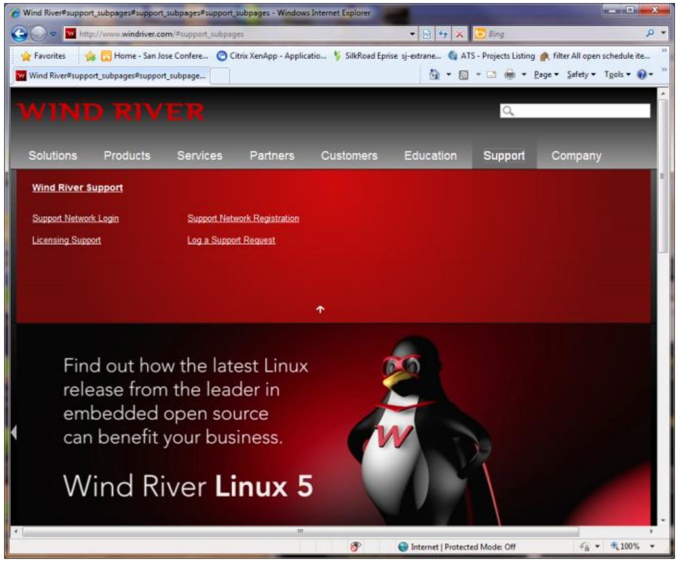
- Current versions
  - Latest stable: refer to linux-socfpga git repo tags: rel_socfpga-x.x…
    - We keep up with Linus Torvalds releases
  - LTSI v3.10
    - Maintained for its lifetime
    - Next LTSI version for SoCFPGA will be 4.1 – available end of 2015
  - Real-Time: 3.10-ltsi-rt
    - LTSI kernel with PREEMP_RT patches
- All branches are kept in sync
  - Bug fixes
  - New features
  - No changes of API in the LTSI branches

Wind River Linux

yocto™ PROJECT COMPATIBLE

- Wind River Linux version 7
- Linux SMP Kernel version 3.14 (LTSI)
  - Real Time patches & Carrier Grade Linux available
- Yocto project user space
- Bitbake build system
- WR Workbench Tools
- Available now from Wind River
- Technical Support:
  - www.windriver.com
  - support@windriver.com

WIND RIVER

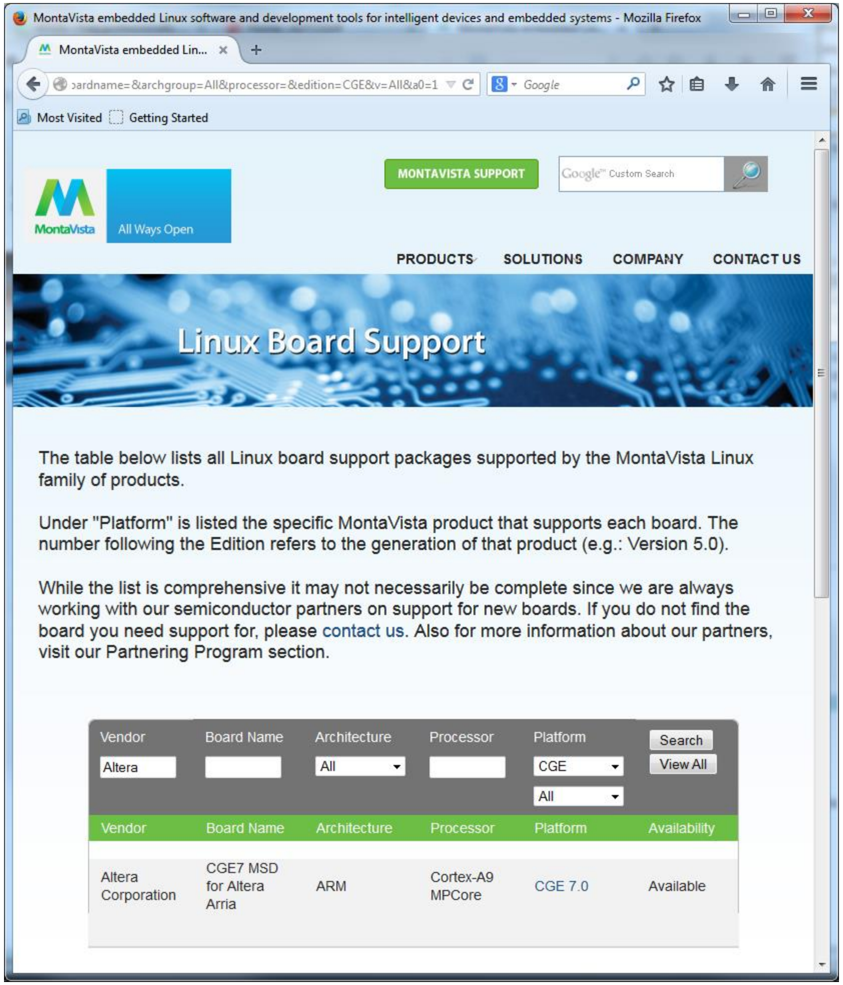

Wind River#support_subpages#support_subpages#support_subpages - Windows Internet Explorer
http://www.windriver.com/#support_subpages
Bing
Favorites
Home - San Jose Confere...
Citrix XenApp - Applicatio...
SilkRoad Eprise sj-extrane...
ATS - Projects Listing
filter All open schedule ite...
Wind River#support_subpages#support_subpage...
Page
Safety
Tools
WIND RIVER
Solutions
Products
Services
Partners
Customers
Education
Support
Company
Wind River Support
Support Network Login
Support Network Registration
Licensing Support
Log a Support Request
Internet | Protected Mode: Off
100%

Find out how the latest Linux release from the leader in embedded open source can benefit your business.

Wind River **Linux 5**

Monta Vista Linux

- Monta Vista Linux CGE7
  - Carrier Grade Edition
- Linux SMP Kernel version 3.10 (LTSI)
- Yocto project user space
- Available from Monta Vista
- Technical Support:
  - www.mvista.com


MontaVista embedded Linux software and development tools for intelligent devices and embedded systems - Mozilla Firefox
MontaVista embedded Lin...
ɔardname=&archgroup=All&processor=&edition=CGE&v=All&a0=1
Google
Most Visited
Getting Started
MontaVista
All Ways Open
MONTAVISTA SUPPORT
Google™ Custom Search
PRODUCTS
SOLUTIONS
COMPANY
CONTACT US

Linux Board Support

The table below lists all Linux board support packages supported by the MontaVista Linux family of products.

Under "Platform" is listed the specific MontaVista product that supports each board. The number following the Edition refers to the generation of that product (e.g.: Version 5.0).

While the list is comprehensive it may not necessarily be complete since we are always working with our semiconductor partners on support for new boards. If you do not find the board you need support for, please contact us. Also for more information about our partners, visit our Partnering Program section.

Vendor: Altera
Board Name:
Architecture: All
Processor:
Platform: CGE, All

Vendor	Board Name	Architecture	Processor	Platform	Availability
Altera Corporation	CGE7 MSD for Altera Arria	ARM	Cortex-A9 MPCore	CGE 7.0	Available

Latest Stable Kernel vs. LTSI Kernel

Latest Stable Kernel

- Access to latest kernel features
- New features and drivers often released only to the latest kernel version
- Significant investment in kernel maintenance:
  - Back porting features, bug fixes, device support, & new drivers or…
  - Constant upgrades to latest stable kernel

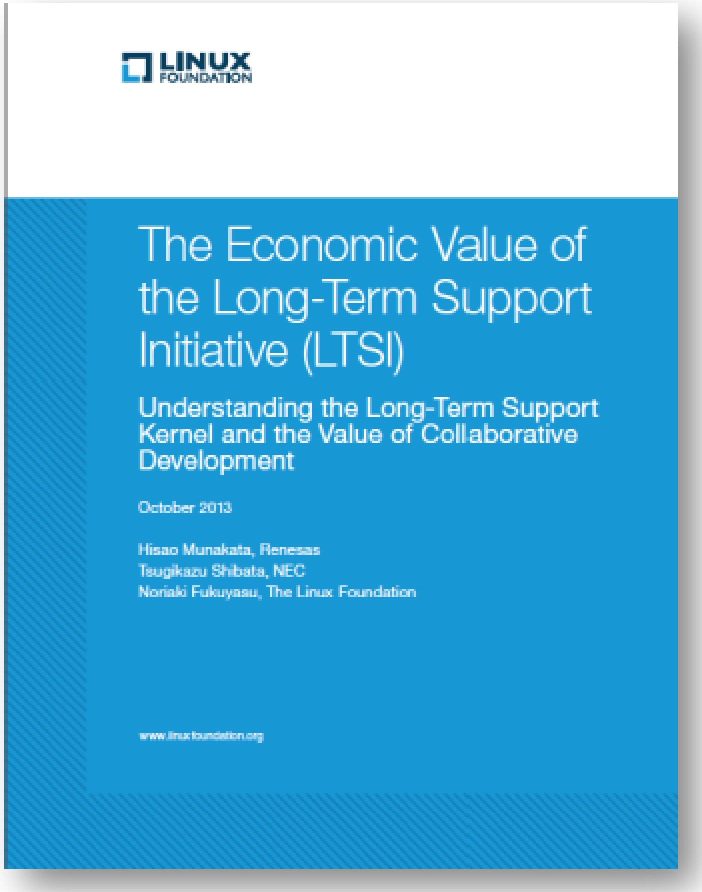
LTSI Kernel

- LTSI kernel versions supported for 2 years
- Critical bug fixes, priority features, & new device or driver support back-ported to LTSI kernel by the community
- Reduces investment in kernel maintenance
  - Features, bug fixes, device support, & new drivers ported by the community

LTSI – Economic value

Economic Value of LTSI?

- Cost of Back porting security and bug fixes are 3M$ per year per version
- Cost of maintaining in-house patch is 288K$ in case of LTSI 3.4



LINUX FOUNDATION

The Economic Value of the Long-Term Support Initiative (LTSI)

Understanding the Long-Term Support Kernel and the Value of Collaborative Development

October 2013

Hisao Munakata, Renesas
Tsugikazu Shibata, NEC
Noriaki Fukuyasu, The Linux Foundation

www.linuxfoundation.org

Linux Code Quality

- Altera’s internal development process is similar to the community’s
  - Code/Peer reviews
  - Code style checked
  - Copyright, licenses, etc checked
  - All checks enforced
- Daily Builds
  - Automated builds run daily
  - Complete system: boot loader, kernel, Angstrom
  - SD card image produced
- Daily Tests
  - Linaro’s LAVA is used
  - All kernel branches tested

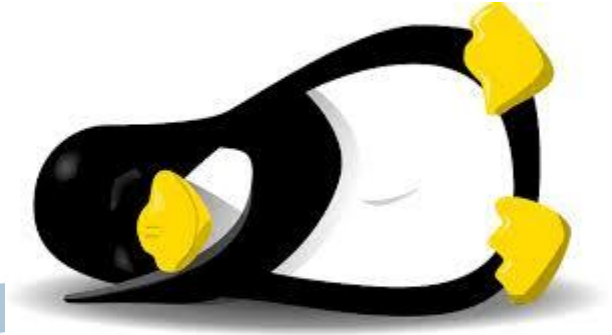
Linux Code Tests

- Objectives
  - Daily test of the supported Linux kernel branches
  - Provide feedback to developers
- Infrastructure
  - Linaro's LAVA is used
    - Linaro Automated Validation Architecture
    - Runs our unit tests and log results
  - Tests start automatically after each build is complete

Altera SoC Linux Support Model

- Rocketboards.org
  - SoC & Nios II Linux documentation
  - SoC & Nios II SoC Linux reference & example designs
- Rocketboards.org RFI & Linux Community
  - Kernel/RFS/u-boot questions
  - SoC/Nios II subsystem and driver questions
- Altera.com and Rocketboards.org
  - SoCEDS & Quartus/QSys documentation and questions
  - SoC Preloader questions
  - SoC HPS implementation specific questions
  - Use myAltera for service requests
- Support from Altera is focused on SoCFPGA and NiosII Linux Board Support Package
- Altera enables Linux community development on SoCFPGA & Nios II

Break



Components of the SoC FPGA Linux BSP

Building a Custom Embedded Linux Distribution

How do I get from here….

…to here?

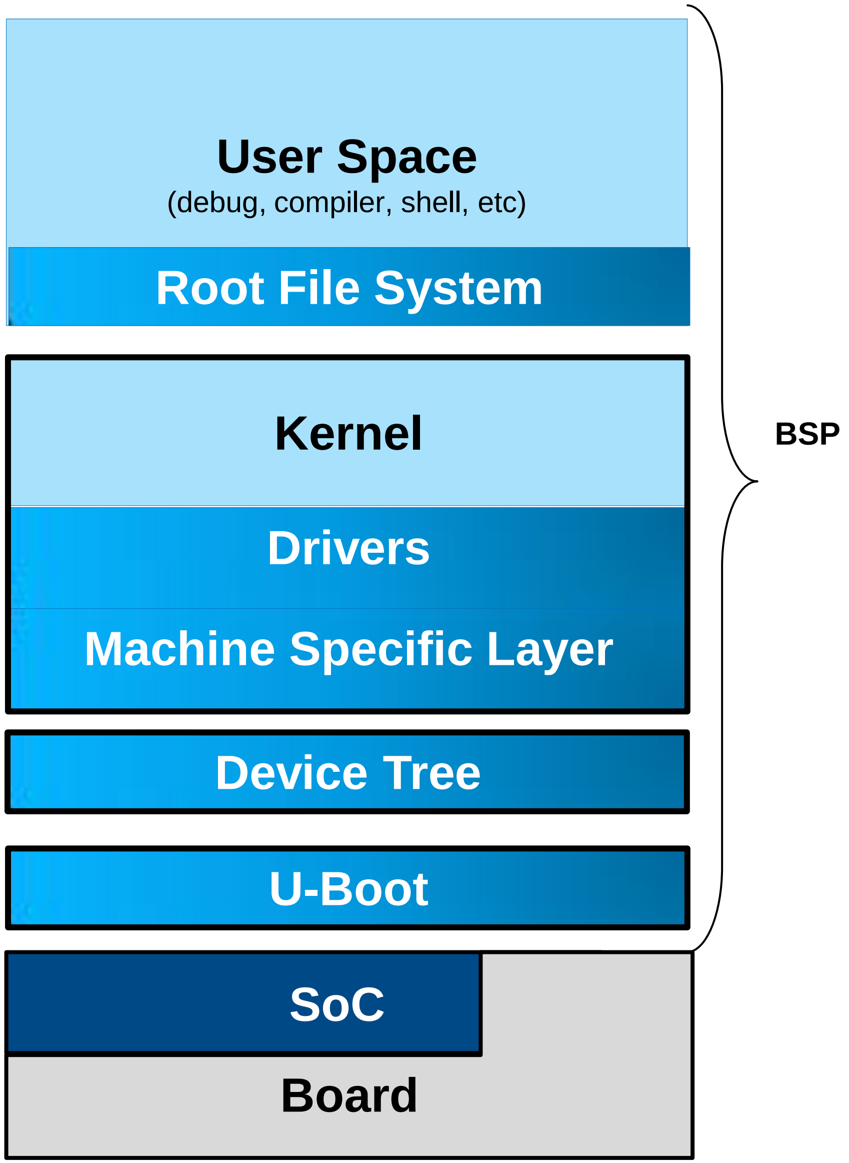
- Altera provides a Linux BSP…

  …not a Linux distribution.
- The BSP enables the creation of a custom distribution

SoC Linux Board Support Package

- U-Boot
- Device Tree
- SoC Machine Specific Layer – mach-socfpga
- Drivers for SoC and board components
- Kernel
- Root File System

SoC Linux BSP release provides all of the components in fully non-proprietary source code form



SoC Linux Board Support Package

Example configuration to enable evaluation and initial development →

Kernel configuration to enable evaluation and initial development →

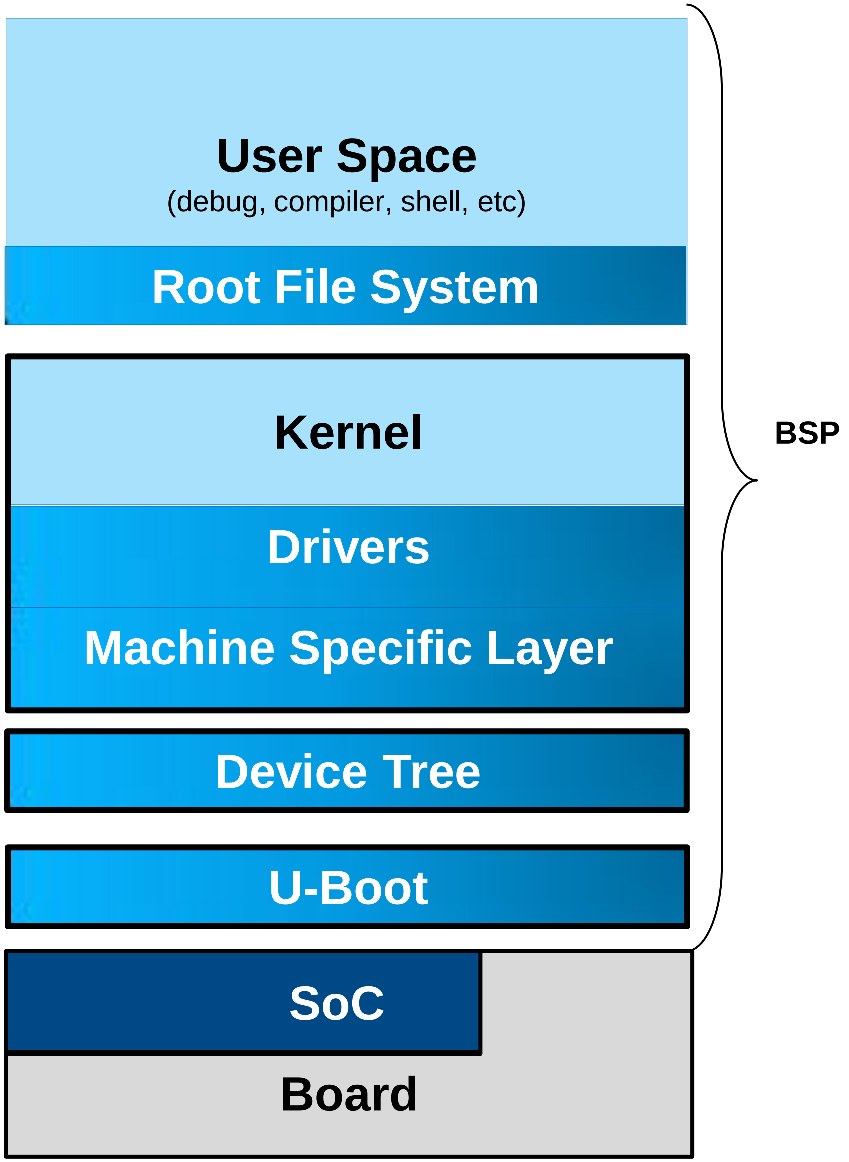
Up-streamed and community supported drivers →

Up-streamed mach-socfpga architecture →

Board-specific layer which enables common kernel binary →

Open-source, community supported boot loader →

Development kit or custom board →



Cyclone V SoC Development Kit

- Everything you need to begin Linux development
  - SoC Development Kit
  - Golden System Reference Design
    - A Linux distribution for the dev. kit
- Features:
  - 1 user license for ARM DS-5 Altera Edition Toolkit
  - Ethernet, USB, CAN, UART
  - DDR3 (HPS and FPGA), SDCard, QSPI
  - PCIe (rootport & endpoint)
  - Expansion header
  - Much more . . .

www.altera.com/products/devkits/altera/kit-cyclone-v-soc.html

Linux GSRD for Development Kits

- Boot Linux from an SD card
  - Updated images on: http://releases.rocketboards.org/
    - Choose a release date folder, the “gsrd” folder, then the “bin” folder. Ex: http://releases.rocketboards.org/release/2014.12/gsrd/bin
  - 3.10 LTSI kernel
  - Angstrom Linux distribution for SoC
    - Package manager to load packages from Angstrom’s on-line package feed
    - Add whatever tools are needed for evaluation: gstreamer, usb-utils, etc…
- GSRD contents
  - Complete HW reference design w/ FPGA programming file
  - Bootable SD card image & component binaries
  - Tagged for rebuilding in angstrom-socfpga git repository ACDSX.X_REL_GSRD_PR

A complete SD card ships with the board

- **Take it out and stick it in a drawer**
- **Based on out-of date 3.9 kernel**

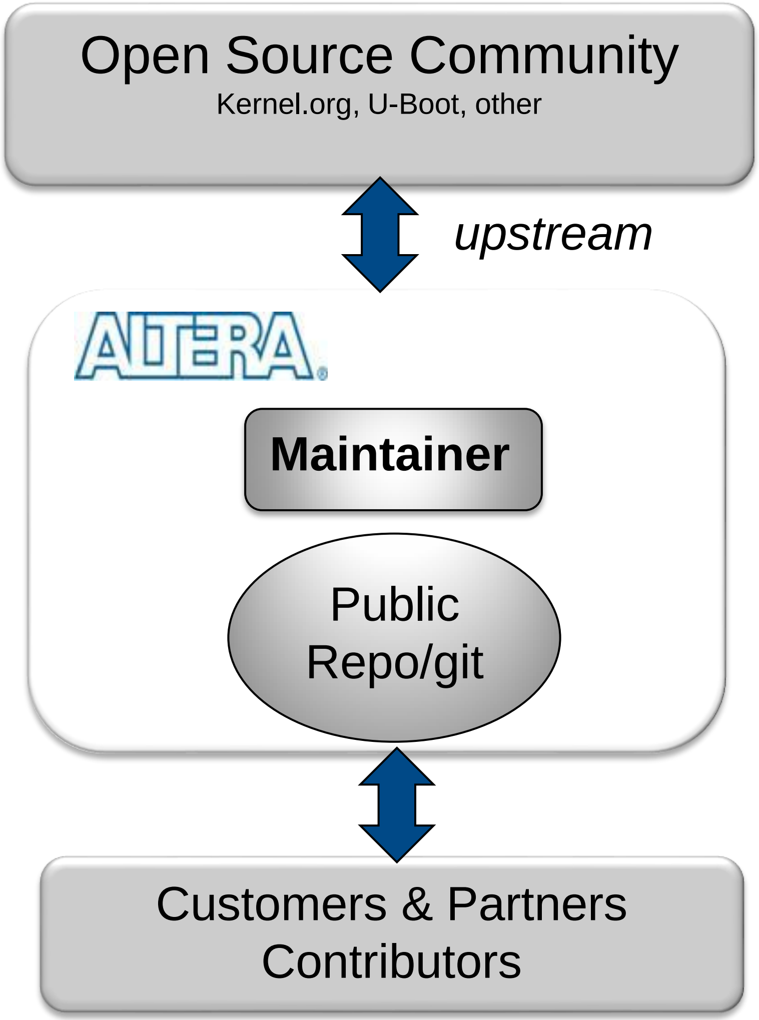
Multiple Dev. Kit Options w/ Linux BSP

Kit	Vendor	Family
Arria 10 SoC Dev. Kit	Altera	**Arria 10 SoC**
Arria V SoC Dev. Kit	Altera	**Arria V SoC**
Cyclone V SoC Dev. Kit	Altera	**Cyclone V SoC**
Atlas Board	Altera	**Cyclone V SoC**
SoCKit	Arrow	**Cyclone V SoC**
Helio SoC Eval. Platform	Macnica	**Cyclone V SoC**
SoCrates	EBV Elektronik	**Cyclone V SoC**

SoC Linux Up-streaming & Driver Support

Maintaining and Up-streaming

- Altera awarded maintainership for the ‘SoC FPGA’ architecture
  - Kernel (arch/arm/mach-socfpga)
  - U-Boot (altera/socfpga_cyclone5)
- Being a maintainer means
  - We upstream the SoC related code
  - We control the changes against the SoC code requested by the community
  - See kernel.org and git.denx.de
- Other community contributions
  - Device Tree Generator: sopc2dts
  - Yocto meta-altera layer



Open Source Community

Kernel.org, U-Boot, other

upstream

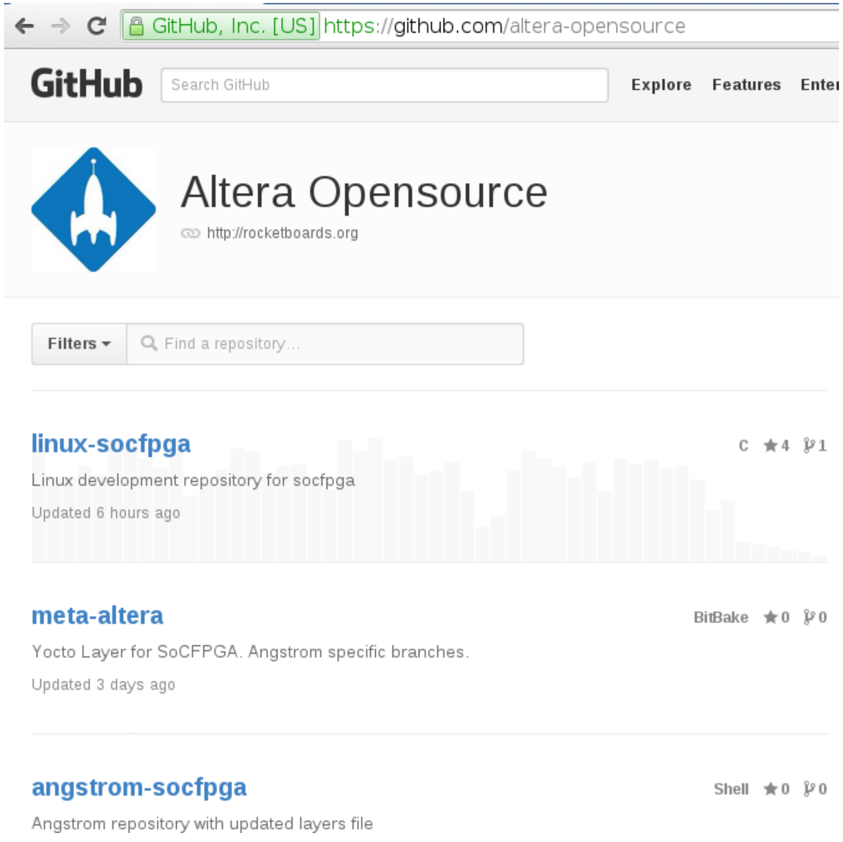
Maintainer

Public Repo/git

Customers & Partners Contributors

SoCFPGA Linux Code Repositories on GitHub

- Public git repos for SoCFPGA
  https://github.com/altera-opensource
- Migrated from RocketBoards
  - **Rocketboards.org git repos are no longer updated**



GitHub, Inc. [US] https://github.com/altera-opensource

Altera Opensource

http://rocketboards.org

linux-socfpga C ★4 1

Linux development repository for socfpga

Updated 6 hours ago

meta-altera BitBake ★0 0

Yocto Layer for SoCFPGA. Angstrom specific branches.

Updated 3 days ago

angstrom-socfpga Shell ★0 0

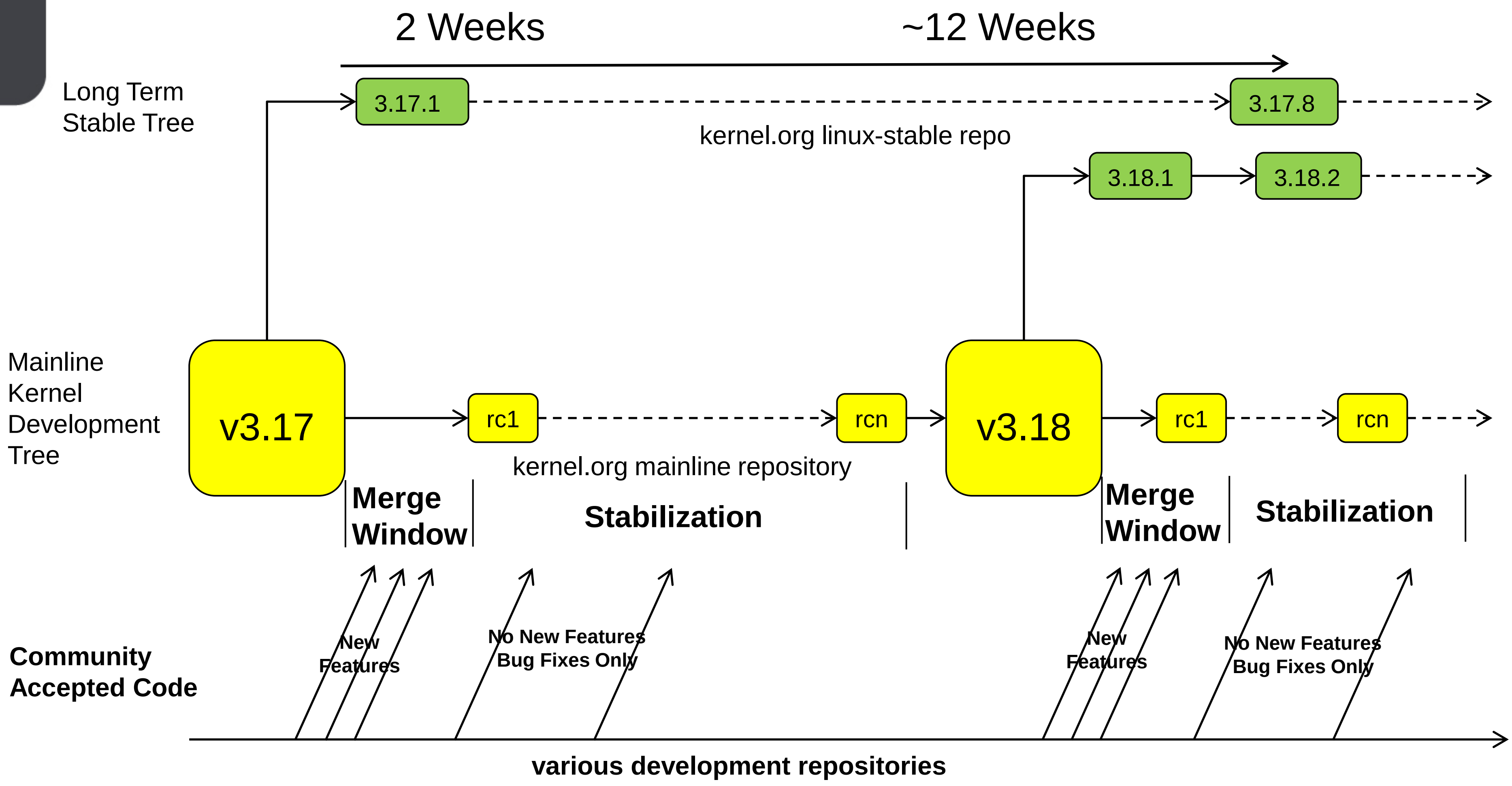
Angstrom repository with updated layers file

SoCFPGA Linux Code Repositories on GitHub

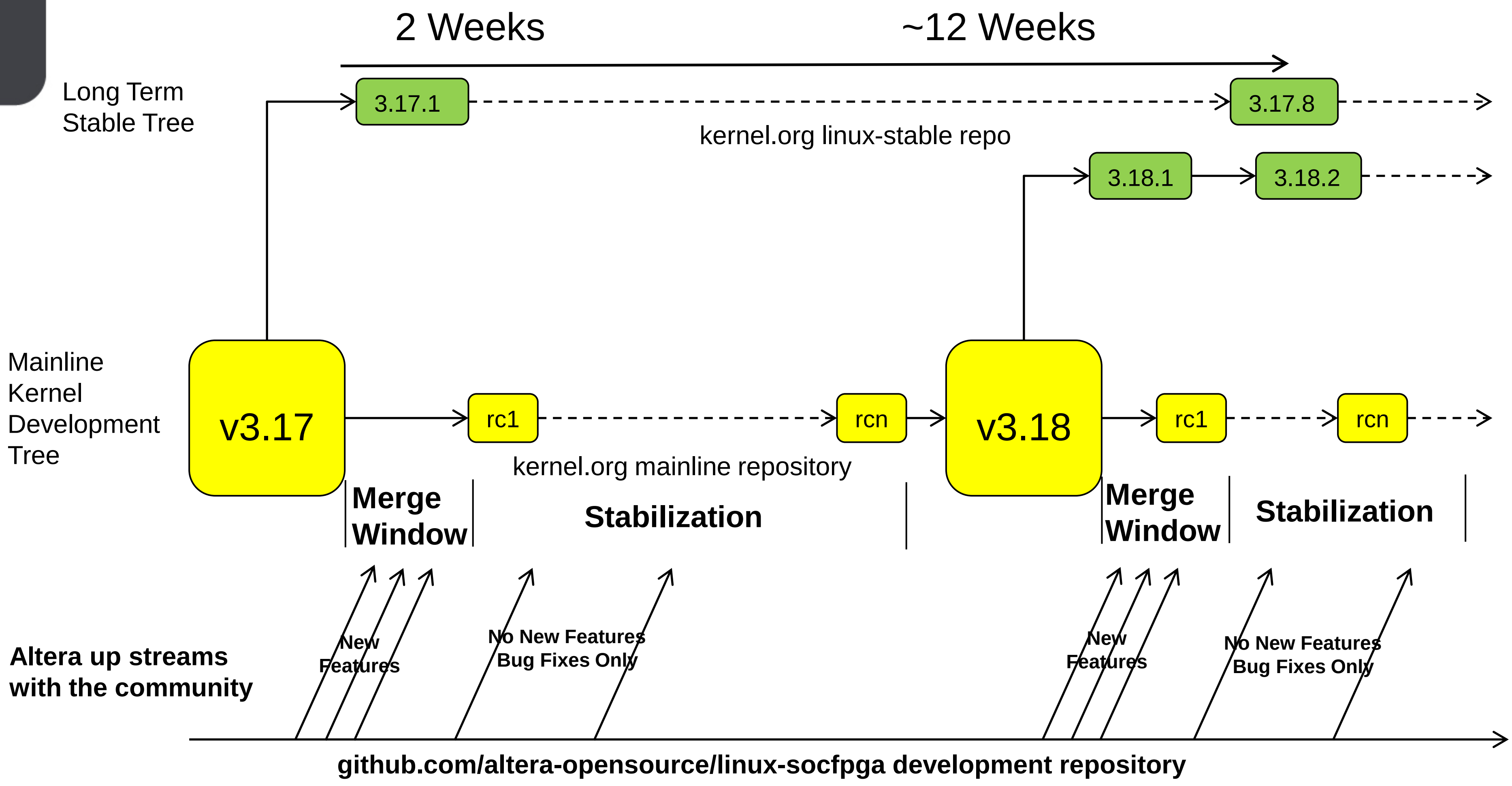
Repository	Description
linux-socfpga.git/	SoCFPGA Linux development repository • Mirrors kernel.org linux repo releases • Downstream branches for socfpga specific patches and updates
meta-altera.git/	Repository for Yocto recipes for SoCFPGA • Starting point for custom Yocto recipes
angstrom-socfpga.git/	Setup scripts for SoCFPGA Angstrom distribution
uboot-socfpga.git/	SoCFPGA u-boot development repository
sopc2dts.git/	Device Tree Generator (sopc2dts) repository
linux-refdesigns.git/	SW source code for Linux reference designs

**** Sourced from: github.com/altera-opensource/***

Kernel Release Cycle – Merge & Bug Fix

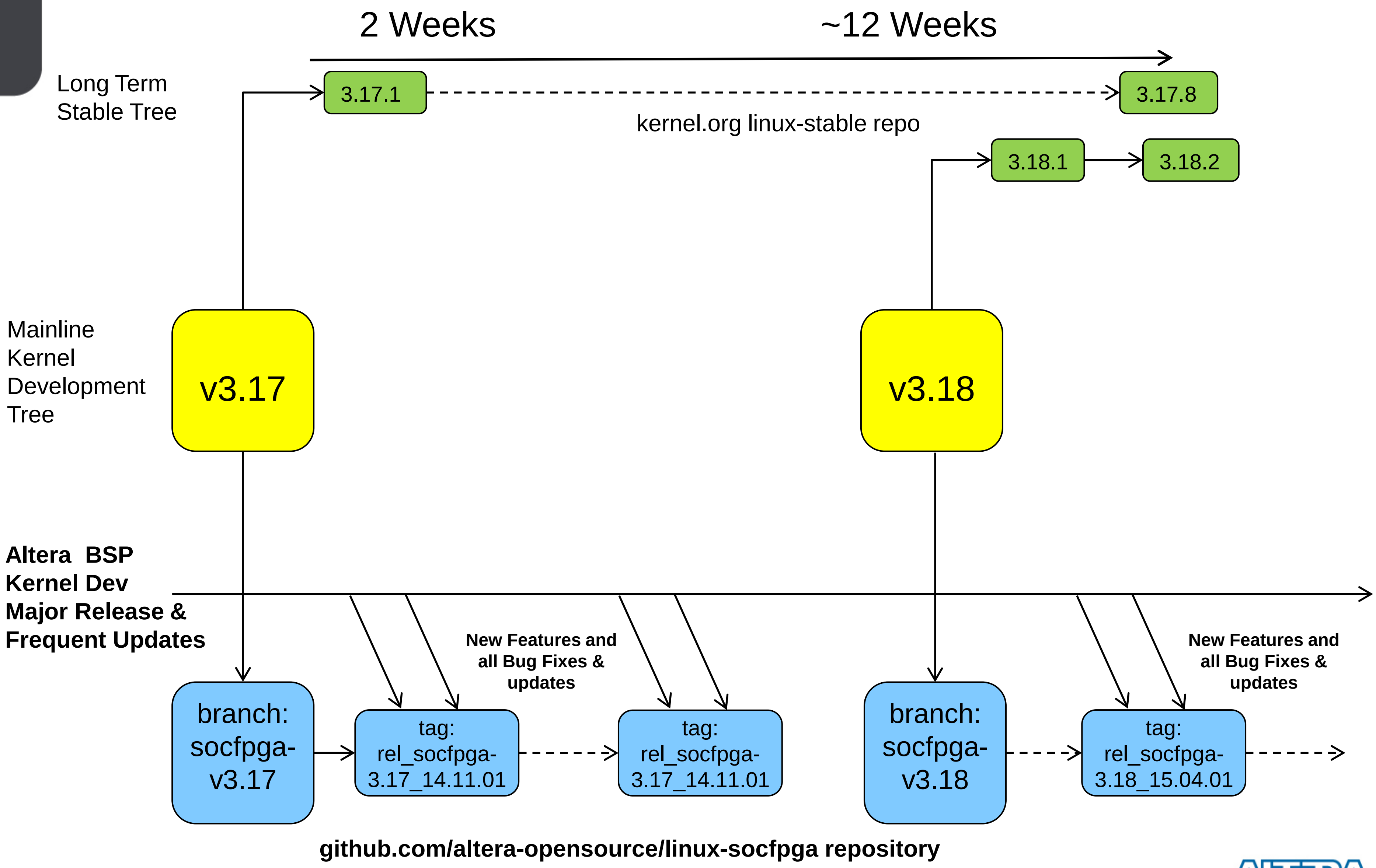


Kernel Release Cycle – Merge & Bug Fix

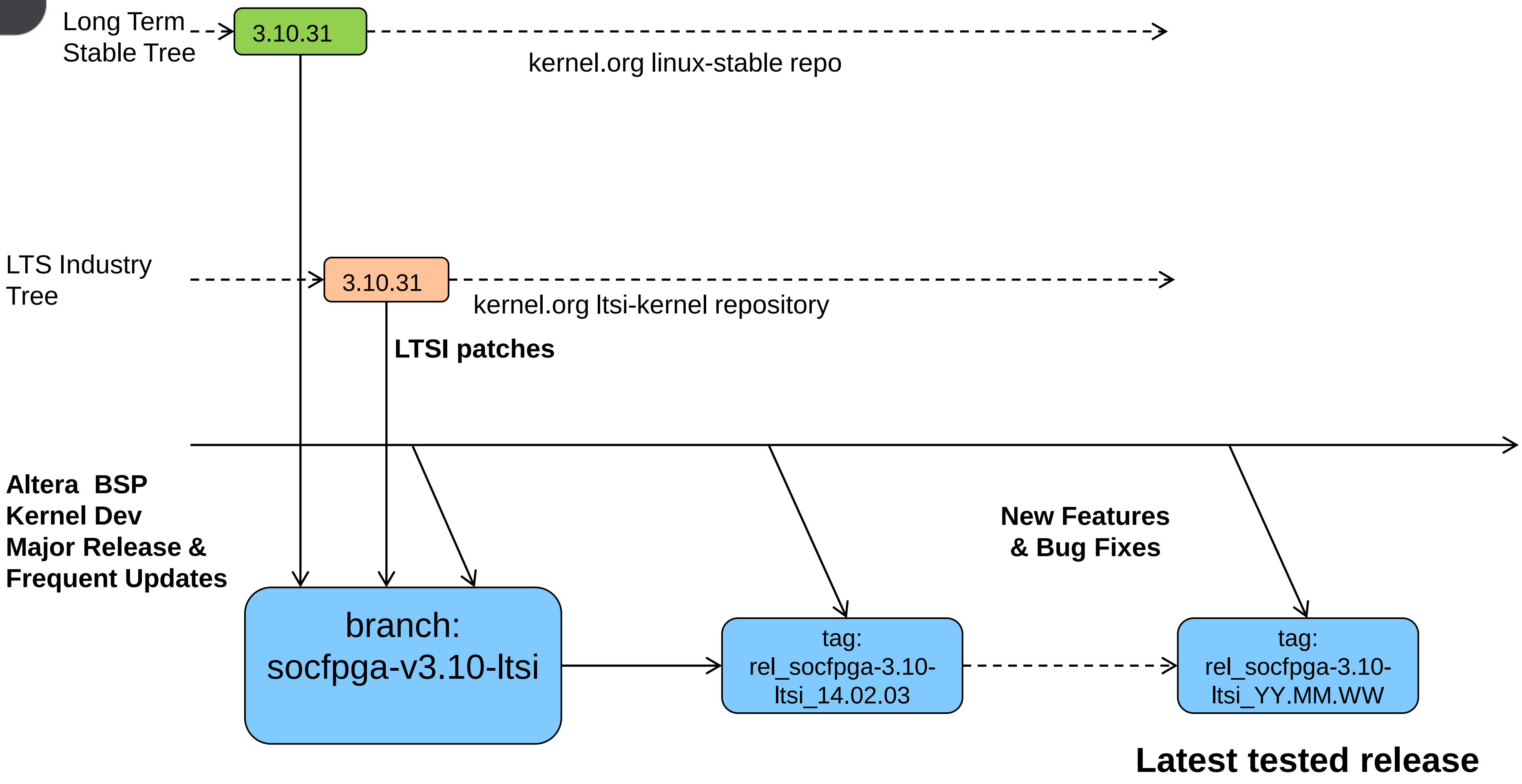


Altera moves with the community to the latest stable kernel

Altera BSP Kernel Development


2 Weeks
~12 Weeks
Long Term Stable Tree
3.17.1
3.17.8
kernel.org linux-stable repo
3.18.1
3.18.2
Mainline Kernel Development Tree
v3.17
v3.18
Altera BSP Kernel Dev Major Release & Frequent Updates
New Features and all Bug Fixes & updates
New Features and all Bug Fixes & updates
branch: socfpga-v3.17
tag: rel_socfpga-3.17_14.11.01
tag: rel_socfpga-3.17_14.11.01
branch: socfpga-v3.18
tag: rel_socfpga-3.18_15.04.01
github.com/altera-opensource/linux-socfpga repository

Altera LTSI Kernel Development



Long Term Stable Tree

3.10.31

kernel.org linux-stable repo

LTS Industry Tree

3.10.31

kernel.org ltsi-kernel repository

LTSI patches

Altera BSP Kernel Dev Major Release & Frequent Updates

branch: socfpga-v3.10-ltsi

tag: rel_socfpga-3.10-ltsi_14.02.03

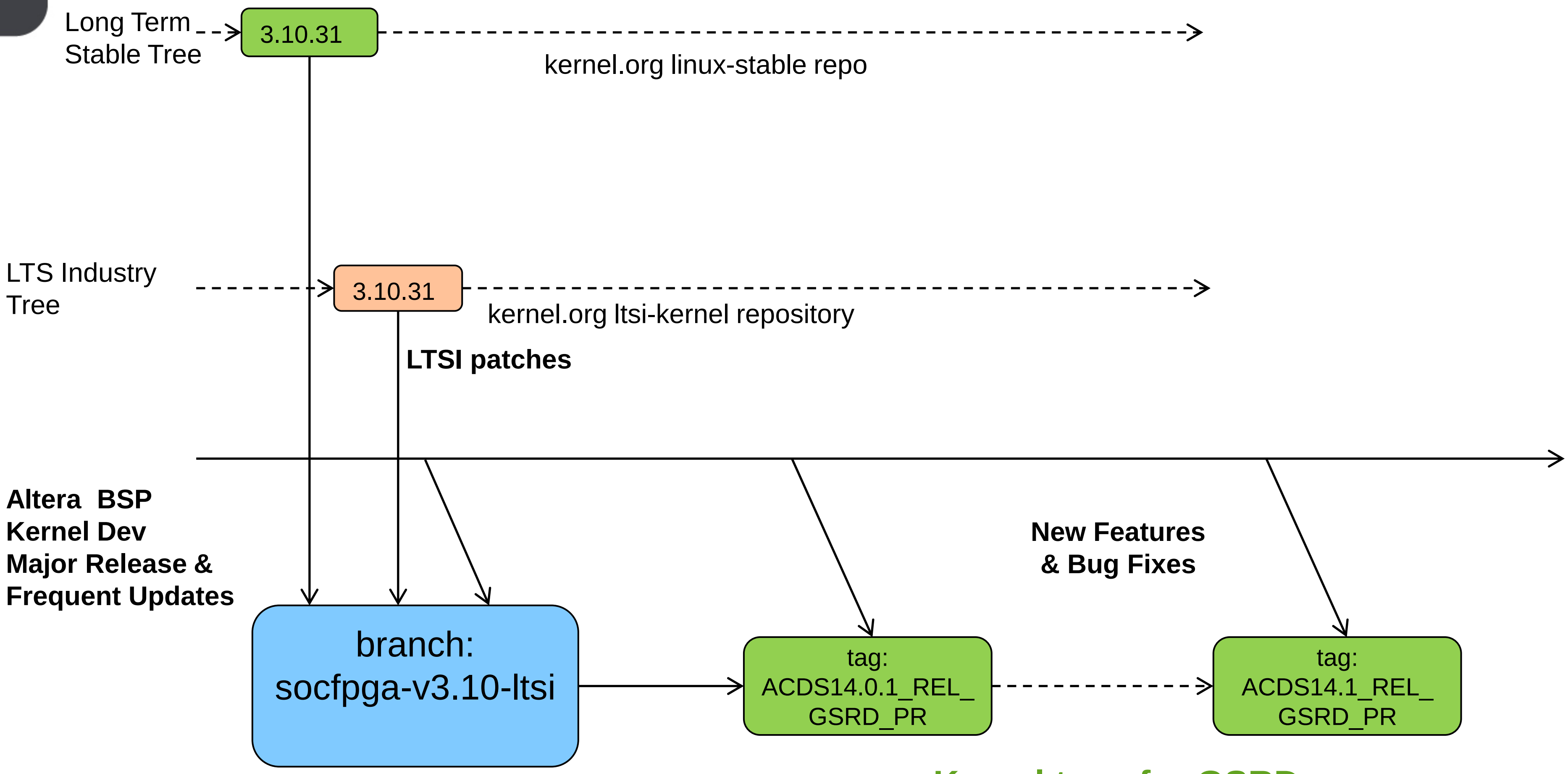
New Features & Bug Fixes

tag: rel_socfpga-3.10-ltsi_YY.MM.WW

Latest tested release

github.com/altera-opensource/linux-socfpga repository

Altera LTSI Kernel Development

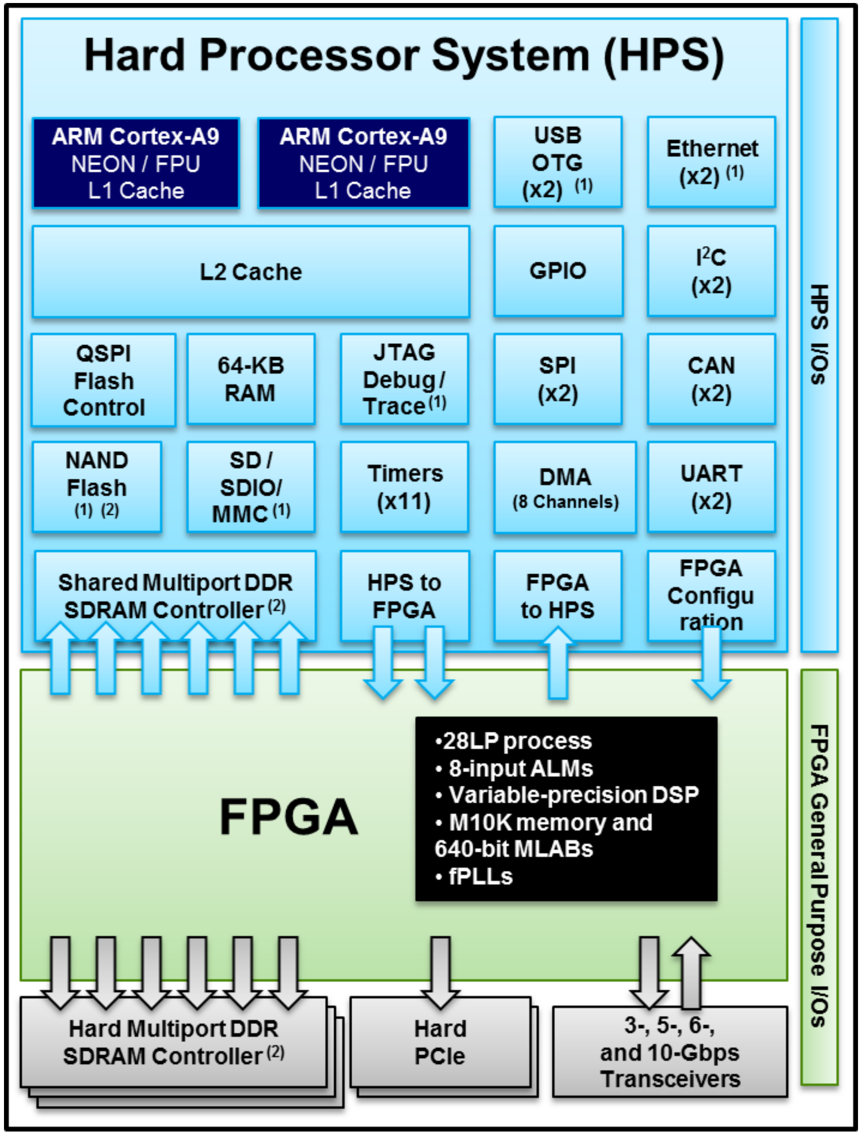


Kernel tags for GSRD releases on Rocketboards

github.com/altera-opensource/linux-socfpga repository

SoC Hard IP Driver Support

- Altera SoC HPS built mainly with off the shelf Hard IP components
  - ARM
  - Synopsys DesignWare
  - Cadence
- Wide-spread usage and community support result in high-quality drivers
- Altera is actively contributing features, updates, and fixes to the community



Notes:
(1) Integrated direct memory access (DMA)
(2) Integrated ECC

Linux Driver Support for HPS Peripherals

Driver	Kernel Source Tree Location	Maintainer
SPI	drivers/spi/spidev.c drivers/spi/spi-dw.c drivers/spi/spi-dw-mmio.c	**Community**
CAN	drivers/net/can/c_can/c_can_platform.c	**Community**
Ethernet	drivers/net/ethernet/stmicro/stmmac/stmmac_platform.c	**Community**
NAND	mtd/nand/denali_dt.c	**Community**
I2C	drivers/i2c/busses/i2c-designware-platdrv.c	**Community**
USB	drivers/usb/dwc2/	**Community**
USB PHY	usb/phy/phy-generic.c	**Community**
SDMMC	drivers/mmc/host/dw_mmc-pltfm.c	**Community**
Timer	drivers/clocksource/dw_apb_timer_of.c	**Community**

Linux Driver Support for HPS Peripherals

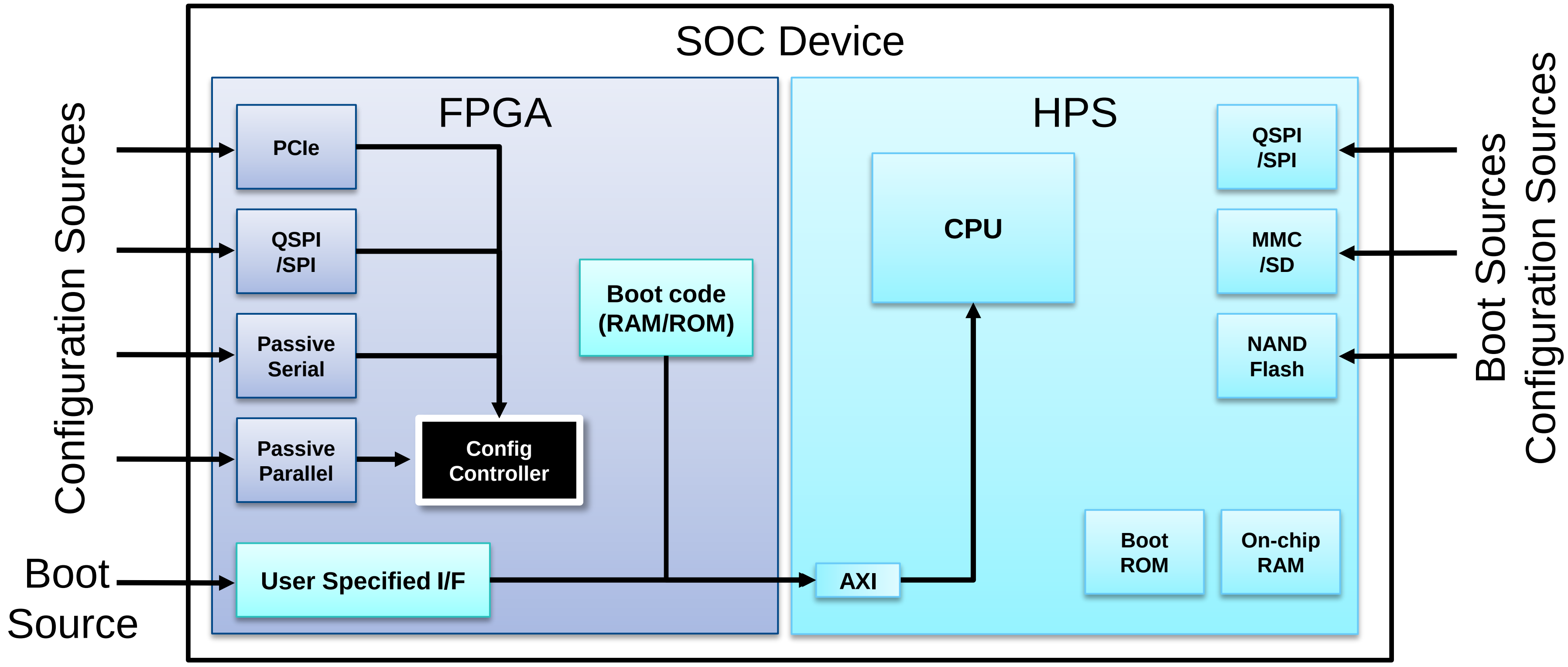
Driver	Kernel Source Tree Location	Maintainer
Watchdog	drivers/watchdog/dw_wdt.c	**Community**
PL330 DMA	dma/pl330.c	**Community**
GIC	drivers/irqchip/irq-gic.c	**Community**
GPIO	drivers/gpio/gpio-dwapb.c	**Community**
Timer	drivers/clocksource/dw_apb_timer_of.c	**Community**
UART	drivers/tty/serial/8250/8250_dw.c	**Community**
QSPI	spi/spi-cadence-qspi.c	**Altera**
Clock Manager	drivers/clk/socfpga/clk.c	**Altera**
FPGA Manager	drivers/fpga/fpga-mgrs/altera.c	**Altera**
FPGA Bridges	drivers/misc/fpga-bridge/	**Altera**
EDAC (ECC)	drivers/edac/altera_*	**Altera**

Linux Driver Support for soft Peripherals

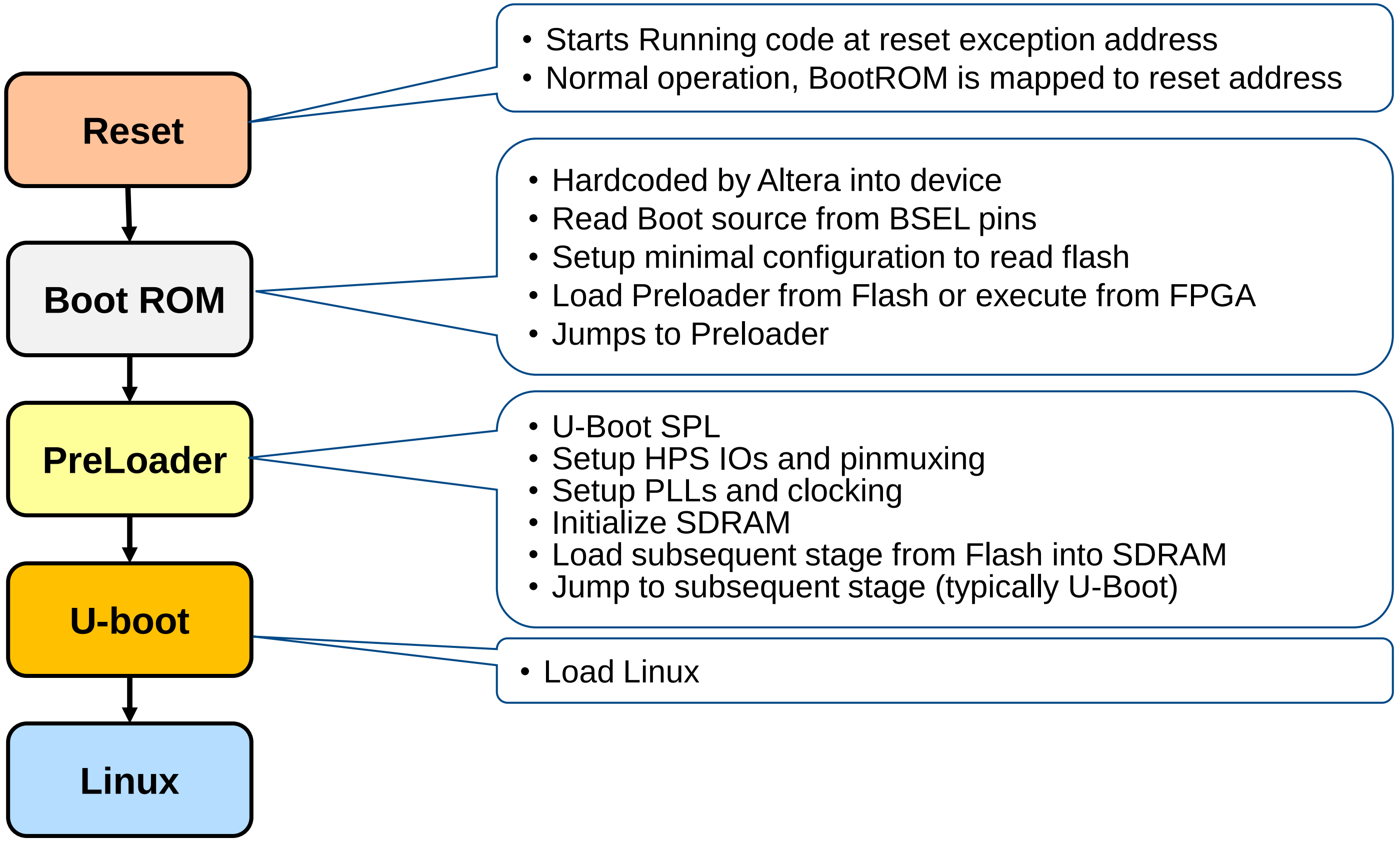
Driver	Kernel Source Tree Location	Maintainer
TSE Ethernet	drivers/net/ethernet/altera	**Altera**
PCIe Root Port	www.rocketboards.org www.rocketboards.org – w/o MSI	**Altera**
Frame Buffer	video/altvipfb.c	**Community**
Avalon SPI	spi/spi-altera.c	**Community**
Avalon UART	tty/serial/altera_uart.c	**Community**
JTAG UART	tty/serial/altera_jtaguart.c	**Community**
QSYS Sys ID	misc/altera_sysid.c	**Altera**
Mailbox	drivers/mailbox/mailbox-altera.c	**Altera**
Altera 16550 UART	drivers/tty/serial/8250/8250_core.c	**Community**
Avalon PIO	drivers/gpio/gpio-altera.c	**Altera**

Altera SoC Linux Boot Flow

Altera SoC FPGA Configuration Options



SoCFPGA Linux Boot Flow – Cyclone V & Arria V SoC



- Starts Running code at reset exception address
- Normal operation, BootROM is mapped to reset address

- Hardcoded by Altera into device
- Read Boot source from BSEL pins
- Setup minimal configuration to read flash
- Load Preloader from Flash or execute from FPGA
- Jumps to Preloader

- U-Boot SPL
- Setup HPS IOs and pinmuxing
- Setup PLLs and clocking
- Initialize SDRAM
- Load subsequent stage from Flash into SDRAM
- Jump to subsequent stage (typically U-Boot)

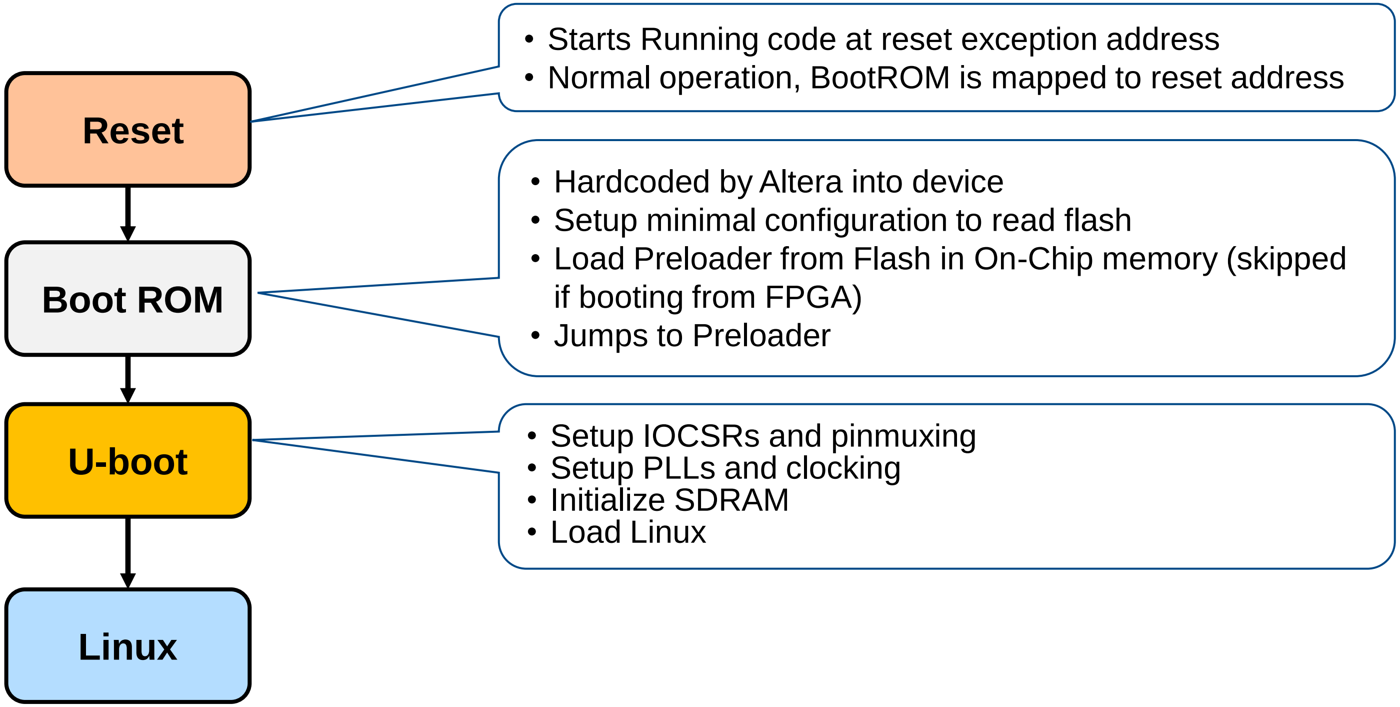
- Load Linux

Preloader Overview – Cyclone V & Arria V SoC

- Loaded by Boot ROM
  - From flash and executed from on-chip RAM
  - or run directly from FPGA
- Uses U-boot Secondary Program Loader (SPL)
  - Open source, GPL Licensed
- Loads U-boot into RAM and jumps to U-boot
- Always regenerate and recompile Preloader when
  - QSys system or HPS configuration changes
  - Quartus/QSys version changes

  Not regenerating and recompiling the Preloader is the single most common source of SoC SW problems!
- Covered in detail in SoCEDS User Guide and *Designing Software for ARM-Base SoC* training class

SoCFPGA Linux Boot Flow – Arria 10



Reset

- Starts Running code at reset exception address
- Normal operation, BootROM is mapped to reset address

Boot ROM

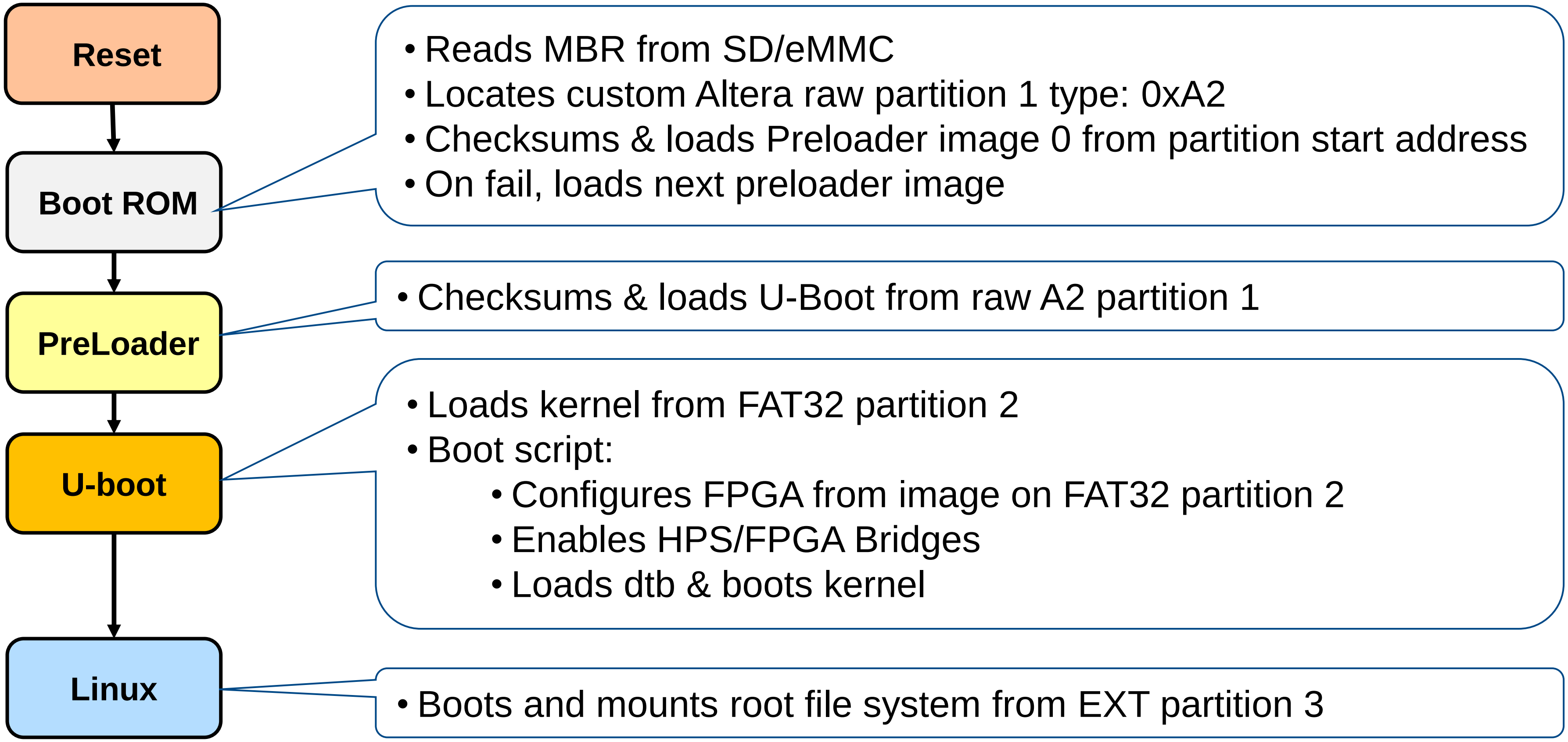
- Hardcoded by Altera into device
- Setup minimal configuration to read flash
- Load Preloader from Flash in On-Chip memory (skipped if booting from FPGA)
- Jumps to Preloader

U-boot

- Setup IOCSRs and pinmuxing
- Setup PLLs and clocking
- Initialize SDRAM
- Load Linux

Linux

Booting from SD/eMMC – GSRD Flow - Simplified



Reset

Boot ROM

- Reads MBR from SD/eMMC
- Locates custom Altera raw partition 1 type: 0xA2
- Checksums & loads Preloader image 0 from partition start address
- On fail, loads next preloader image

PreLoader

- Checksums & loads U-Boot from raw A2 partition 1

U-boot

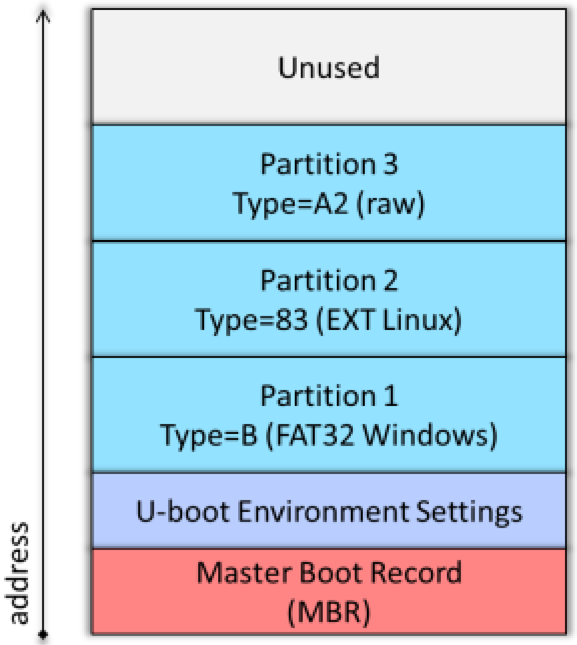
- Loads kernel from FAT32 partition 2
- Boot script:
  - Configures FPGA from image on FAT32 partition 2
  - Enables HPS/FPGA Bridges
  - Loads dtb & boots kernel

Linux

- Boots and mounts root file system from EXT partition 3

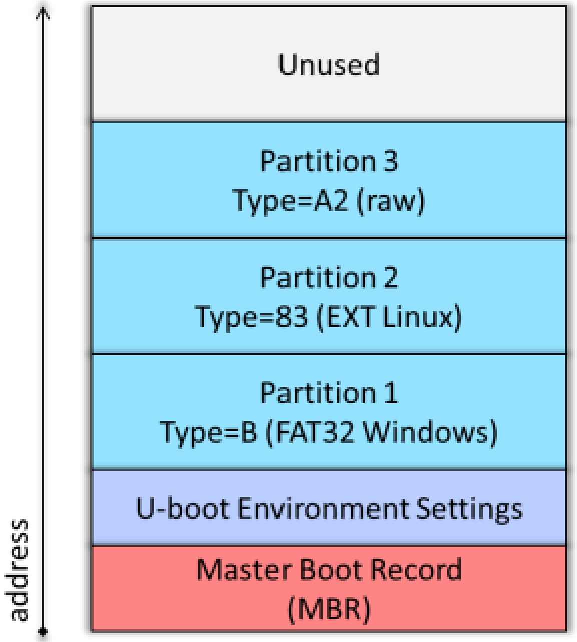
Provides ONE example of an SD/eMMC Linux boot flow

GSRD SD Card Image for Cyclone V & Arria V



Location	File Name	Desciption
Partition 1 (FAT32)	socfpga.dtb	Device Tree Blob
	soc_system.rbf	FPGA configuration file
	u-boot.scr	U-Boot script: configures FPGA and loads kernel
	zImage	Compressed Linux kernel image file
Partition 2 (EXT3)	Various	Linux root file system
Partition 3 (A2 raw)	n/a	Preloader image(s)
	n/a	U-Boot image

GSRD SD Card Image for Arria 10



Location	File Name	Desciption
Partition 1 (FAT32)	socfpga.dtb	Device Tree Blob
	soc_system.rbf	FPGA configuration file
	u-boot.scr	U-Boot script: configures FPGA and loads kernel
	zImage	Compressed Linux kernel image file
Partition 2 (EXT3)	Various	Linux root file system
Partition 3 (A2 raw)	n/a	U-Boot image

Creating SD Card Images

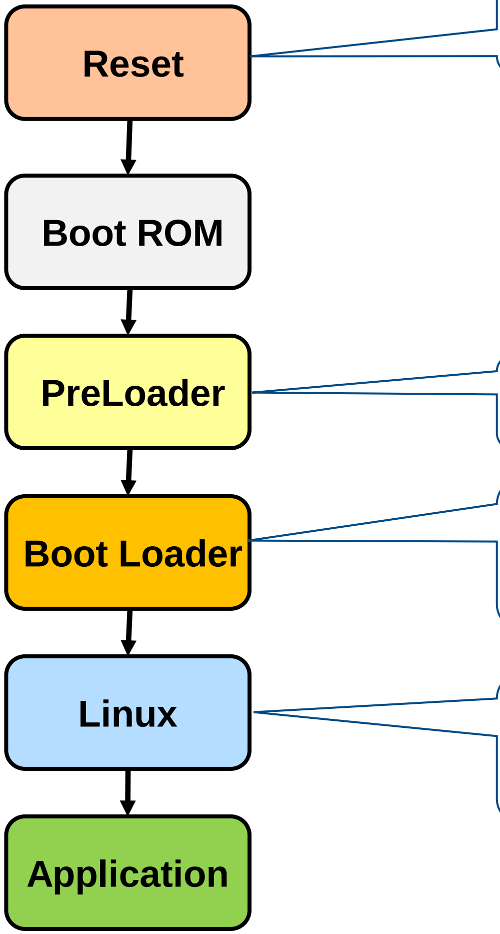
- Create using Altera provided script
  - See “tools” folder under GSRD release folders

    http://releases.rocketboards.org/
  - Builds complete SD card image which can be directly copied
- Use pre-built images
  - In “bin” folder under GSRD release folders

    http://releases.rocketboards.org/
  - <SoCEDS install directory>/examples/software
  - Can be directly copied to SD card
- Described in “Creating and Updating SD Card” section of GSRD User Manual

  http://www.rocketboards.org/foswiki/Documentation/GSRD

FPGA Configuration – Cyclone V & Arria V SoC



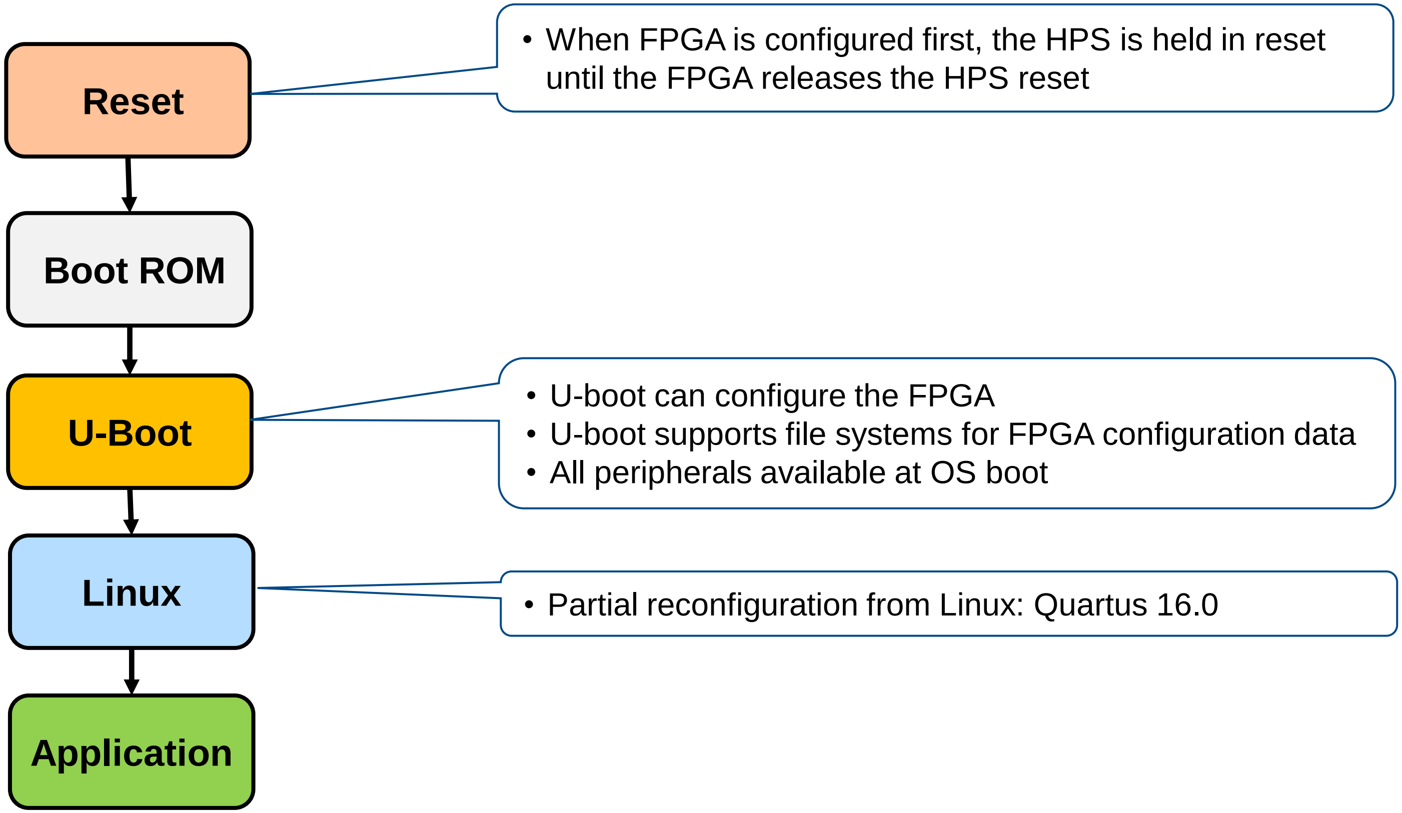
- When FPGA is configured first, the HPS is held in reset until the FPGA releases the HPS reset

- Preloader can configure the FPGA
- Currently requires boot from QSPI

- U-boot can configure the FPGA - recommended for Linux
- U-boot supports file systems for FPGA configuration data
- All peripherals available at OS boot

- Linux can configure the FPGA, but tricky
- Not all peripherals are available at boot
- 28nm F2S Bridge Errata can lock the device

FPGA Configuration – Arria 10 SoC



- When FPGA is configured first, the HPS is held in reset until the FPGA releases the HPS reset

- U-boot can configure the FPGA
- U-boot supports file systems for FPGA configuration data
- All peripherals available at OS boot

- Partial reconfiguration from Linux: Quartus 16.0

Das U-Boot Bootloader

What is u-boot?

- Common embedded bootloader
  - Command line interface w/ decent help and lots of hardware support
  - Capable board bring up tool
  - Driver support for a wide variety of essential peripherals
- Loads the Device Tree and modifies Device Tree configuration at run-time
- Loads the kernel and passes boot arguments
  - From local file system, over network, or over a serial link
- Open-source & GPL licensed
  - http://www.denx.de/wiki/U-Boot

U-Boot for SoCFPGA

- Sourced from GitHub Altera Opensource or SoCEDS

  https://github.com/altera-opensource/u-boot-socfpga.git

  <SoC EDS install dir>/examples/hardware/<*_ghrd>/…/uboot-socfpga
- Supported u-boot versions:
  - Cyclone V SoC & Arria V SoC: 2013.01.01
    - Preliminary support for 2015.xx up streamed to Denx GIT repo
  - Arria 10 SoC: 2014.10
    - 2014.10 & later up streamed to Denx Git repo
- u-boot-socfpga branch & tag convention similar to linux-socfpga

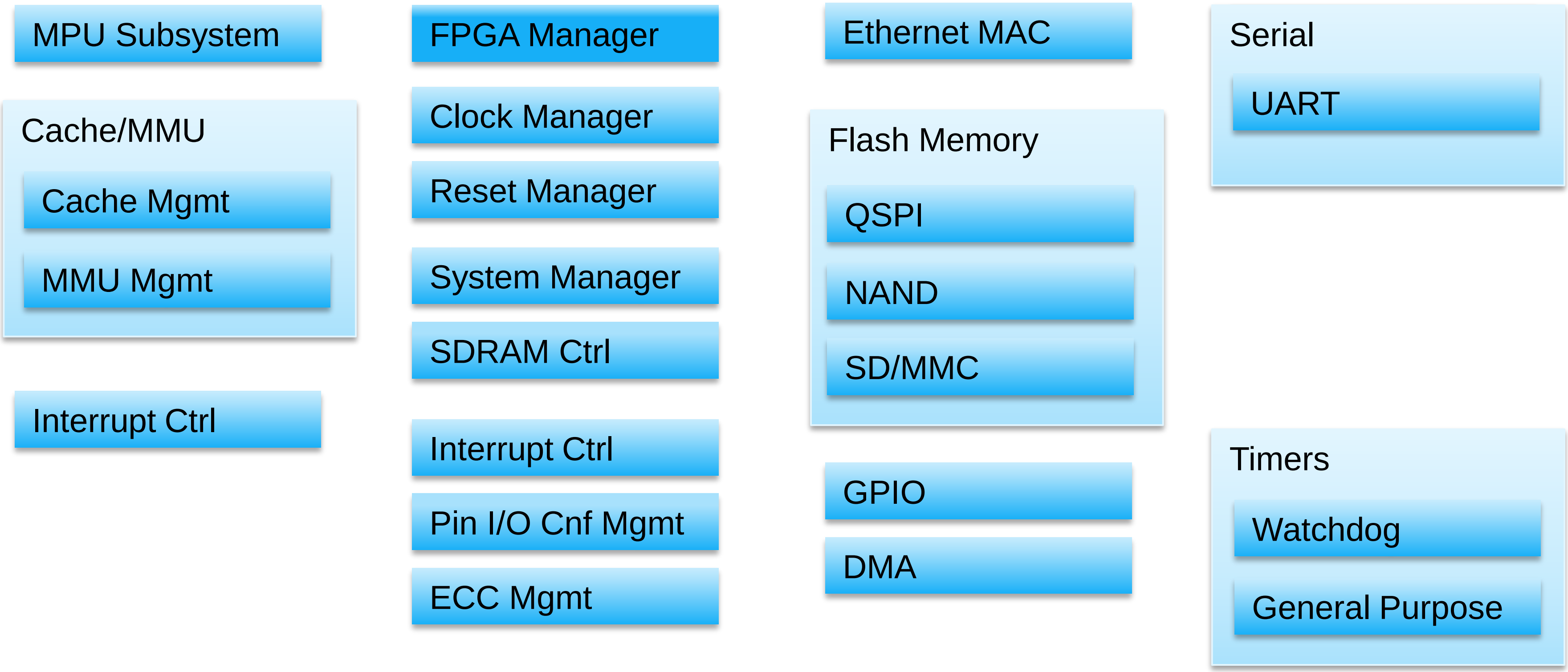
U-Boot for SoCFPGA

- SoCFPGA-specific u-boot documentation

  <u-boot-socfpga repo>/doc/README.SOCFPGA
- u-boot controls SDRAM size passed to kernel
  - u-boot SDRAM sizing algorithm overwrites device tree SDRAM *memory* node entry
  - Device tree entry overrides boot arguments
- SoCFPGA u-boot environment variables
  - Pass HPS/FPGA bridge status and configuration info from preloader
  - Enable FPGA programming from preloader

SoCFPGA HPS Peripherals Supported in U-boot

- Peripheral support in u-boot-socfpga 2013.01.01



MPU Subsystem

Cache/MMU
- Cache Mgmt
- MMU Mgmt

Interrupt Ctrl

FPGA Manager

Clock Manager

Reset Manager

System Manager

SDRAM Ctrl

Interrupt Ctrl

Pin I/O Cnf Mgmt

ECC Mgmt

Ethernet MAC

Flash Memory
- QSPI
- NAND
- SD/MMC

GPIO

DMA

Serial
- UART

Timers
- Watchdog
- General Purpose

Useful U-boot Commands and Variables

- Write memory location

```
mw <address> <size> <optional count>
mw 0xC0000000 0x10 0x6
```

- Read memory location

```
md<optional .b .w .l> <address> <optional size>
md.w 0xC0000000 0x2
```

- Control u-boot auto-boot
  - Boot without delay:

```
setenv bootdelay 0
```

  - Disable auto-boot (stop at u-boot command line):

```
setenv bootdelay -1
```

- Save environment variables to flash

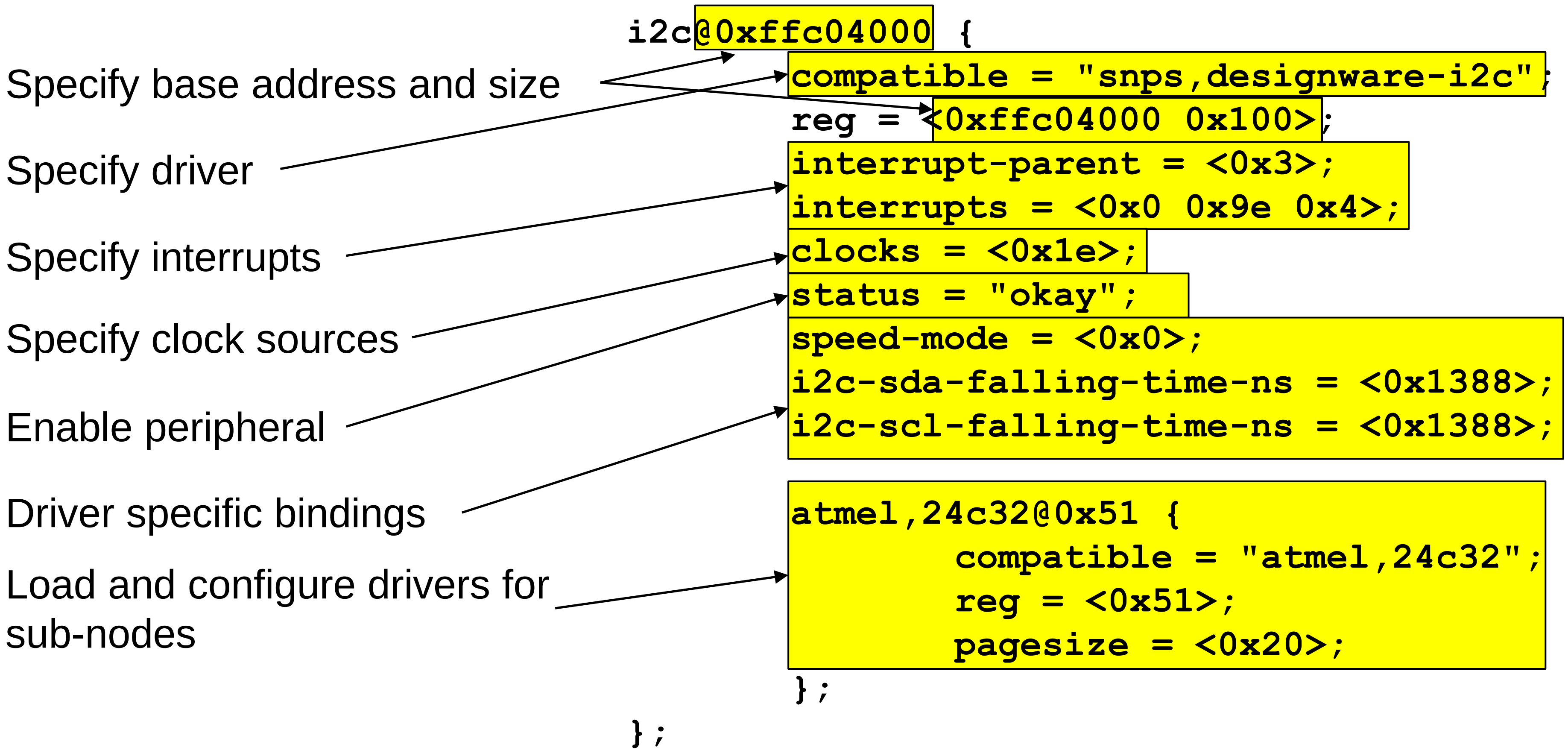
```
saveenv
```

Linux Device Tree for SoC FPGA

What’s a Linux Device Tree?

- A tree-like data structure for describing hardware in embedded systems
- Enables device drivers to be linked to Linux kernel at run-time
  - No Linux kernel recompile required
  - Drivers loaded dynamically after loading Device Tree
- Driver-specific Device Tree bindings are documented in the kernel documentation
  - Documentation/devicetree/bindings
- See Device Tree Generator User Guide
  - See link in GSRD User Guide

SoC FPGA Device Tree Bindings Example



Specify base address and size

Specify driver

Specify interrupts

Specify clock sources

Enable peripheral

Driver specific bindings

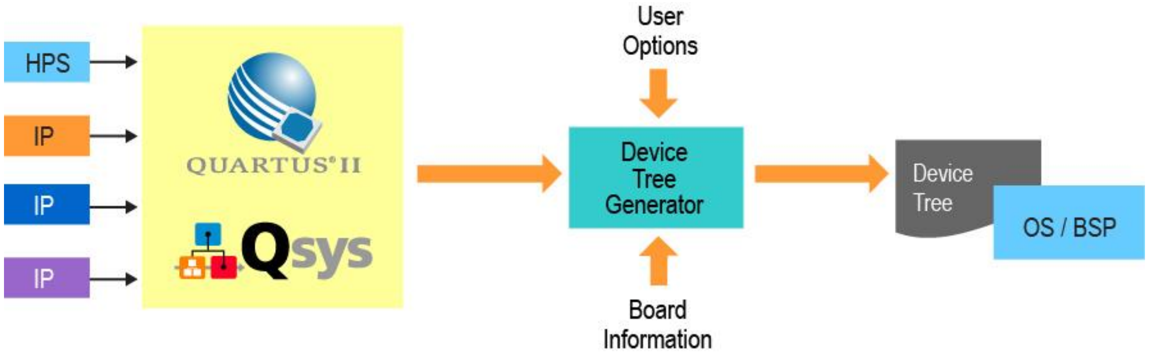
Load and configure drivers for sub-nodes

```
i2c@0xffc04000 {
        compatible = "snps,designware-i2c";
        reg = <0xffc04000 0x100>;
        interrupt-parent = <0x3>;
        interrupts = <0x0 0x9e 0x4>;
        clocks = <0x1e>;
        status = "okay";
        speed-mode = <0x0>;
        i2c-sda-falling-time-ns = <0x1388>;
        i2c-scl-falling-time-ns = <0x1388>;

        atmel,24c32@0x51 {
                compatible = "atmel,24c32";
                reg = <0x51>;
                pagesize = <0x20>;
        };
};
```

Device Trees for a Configurable Peripheral Set

- Typically developers build Linux for fixed form chips
- How do you build Linux for an FPGA fabric that changes?

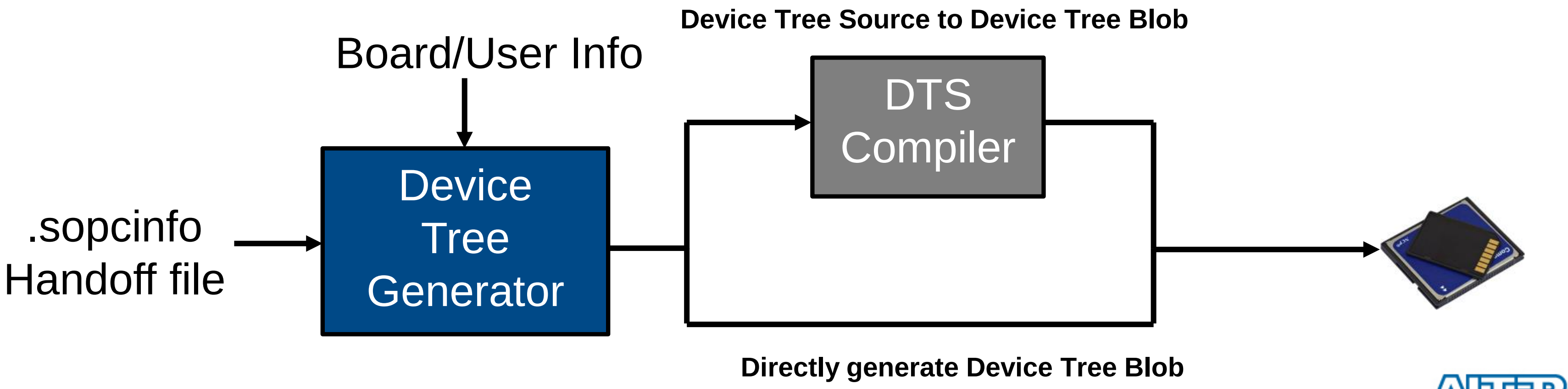
You use the *Device Tree Generator*



- Device tree information can be added to custom Qsys peripherals and custom drivers
- Device Tree Source can be edited by hand

How to use Altera SoC Device Tree Generator

- .sopcinfo file describes HPS and FPGA system
- Board info file describes external devices on board
- Device Tree Generator creates a plain text representation of the device tree – called the Device Tree Source (DTS)
  - Device Tree Generator: SoCEDS sopc2dts tool
  - sopc2dts distributed as part or SoCEDS or on RBO GIT repo
- Compile the text into a binary representation called the Device Tree Blob (DTB)
- Optionally directly generate DTB from sopc2dts


Device Tree Source to Device Tree Blob
Board/User Info
DTS Compiler
.sopcinfo Handoff file
Device Tree Generator
Directly generate Device Tree Blob

Board Info File

- XML file required as an input to sop2dts
- Specifies information of which QSys isn’t aware
  - SPI or I2C timing or external peripherals and their properties
  - External flash (QSPI, SPI, or NAND) properties (organization, specs, etc)
- Board specific Ethernet & PHY information
- Allows the developer to disable peripherals which may be enabled in preloader or FPGA HW

Take Home Lab

Workshop 2 Lab Overview

- Goal:
  - Familiarize you with SoC FPGA Linux components and where to obtain them
- Overview
  - You will build the SoC-specific pieces of a Cyclone V or Arria V SoC Linux distribution and run it on your dev. kit
  - This flow builds each component discretely without a build system. It does not use the optional Yocto or Angstrom based build systems.
  - It configures the distribution in a way which works for this lab, which may differ from your actual system requirements

What You'll Need

- A supported Dev. Kit
  - Altera Atlas Board
  - Altera Cyclone V SoC Board
  - Altera Arria V SoC Board
  - Arrow SoCKIT
  - Macnica Helio Board

Download DE0-Nano-SoC CD from http://cd-de0-nano-soc.terasic.com

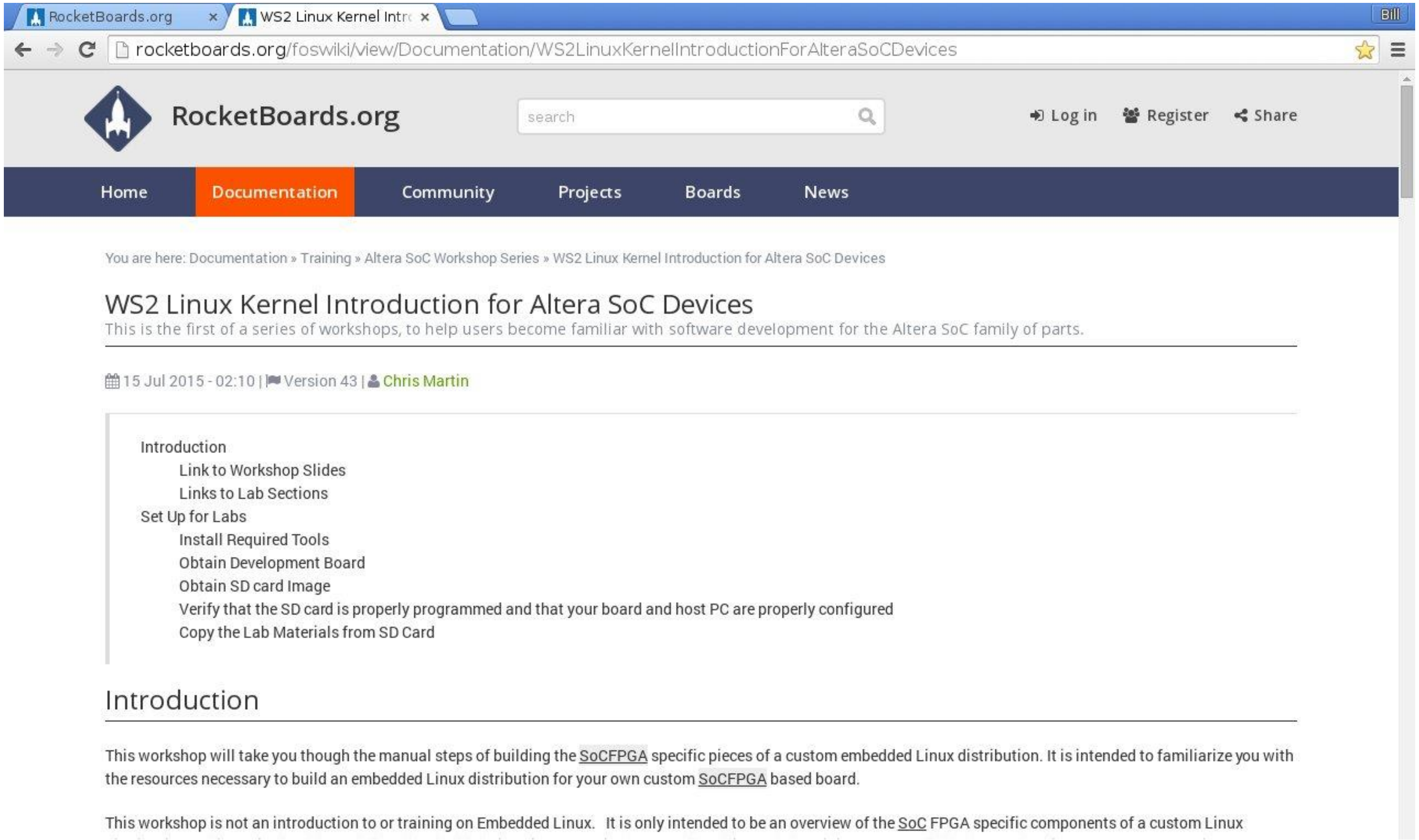
- microSD Card

- Linux machine native or VM
  - 4GByte RAM minimum

- Serial Terminal Application

Obtaining Lab Files and Instructions

- Posted on RocketBoards
  - Link will be emailed out after class



You are here: Documentation » Training » Altera SoC Workshop Series » WS2 Linux Kernel Introduction for Altera SoC Devices

WS2 Linux Kernel Introduction for Altera SoC Devices

This is the first of a series of workshops, to help users become familiar with software development for the Altera SoC family of parts.

15 Jul 2015 - 02:10 | Version 43 | Chris Martin

- Introduction
  - Link to Workshop Slides
  - Links to Lab Sections
- Set Up for Labs
  - Install Required Tools
  - Obtain Development Board
  - Obtain SD card Image
  - Verify that the SD card is properly programmed and that your board and host PC are properly configured
  - Copy the Lab Materials from SD Card

Introduction

This workshop will take you though the manual steps of building the SoCFPGA specific pieces of a custom embedded Linux distribution. It is intended to familiarize you with the resources necessary to build an embedded Linux distribution for your own custom SoCFPGA based board.

This workshop is not an introduction to or training on Embedded Linux. It is only intended to be an overview of the SoC FPGA specific components of a custom Linux

What You Will Accomplish

- Generate and build the preloader
- Generate and compile the device tree
- Obtain and build u-boot
- Obtain, configure, and build the kernel
- Program to an SD card
- Run a Simple Linux App
  - Verifies the lab was completed
- Submit results & feedback

Thank You